D6.8 – Solutions Handbook

WP6 – EVALUTE: Piloting and Demonstrating

i4Q has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 958205.

Document Information

GRANT AGREEMENT NUMBER	958205	**ACRONYM**	i4Q	
FULL TITLE	Industrial Data Services for Quality Control in Smart Manufacturing			
START DATE	01-01-2021	DURATION	41 months	
PROJECT URL	https://www.i4q-project.eu/			
DELIVERABLE	D6.8 – Solutions Handbook v1			
WORK PACKAGE	WP6 – EVALUATE: Piloting and Demonstrating			
DATE OF DELIVERY	CONTRACTUAL	30-Apr-2024	ACTUAL	30-Apr-2024
NATURE	Report	DISSEMINATION LEVEL	Public	
LEAD BENEFICIARY	ITI			
RESPONSIBLE AUTHOR	ITI			
CONTRIBUTIONS FROM	1 - CERTH, 2 - ENG, 3 - IBM, 4 - ITI, 5 - KBZ, 6 - EXOS, 7 - IKER, 8 - BIBA, 9 - UPV, 10 - TUB, 11 - UNINOVA, 19 – WHIRLPOOL, 20 -BIESSE, 21 - FACTOR, 22 – RIASTONE, 23 - FARPLAS, 24 - FIDIA,			
TARGET AUDIENCE	1) i4Q Project partners; 2) industrial community; 3) other H2020 funded projects; 4) scientific community			
DELIVERABLE CONTEXT/ DEPENDENCIES	This document has no preceding documents or further iterations but depends on WP3, WP4 and WP5 results.			
EXTERNAL ANNEXES/ SUPPORTING DOCUMENTS	None			
READING NOTES	None			
ABSTRACT	This deliverable provides a general analysis of the different i4Q Solutions, so that anyone without much knowledge of the project can understand what each of the solutions is and what actions can be carried out with each one of them. The professional profiles are addressed for one each of the i4Q Solutions, so that companies know what type of personnel should be assigned to manage them. The hardware and software requirements necessary to be able to deploy each one of the i4Q Solutions are indicated. The installation and configuration of each one of the i4Q Solutions are also detailed, so that anyone can follow the steps and deploy them.			

A user's manual explaining step-by-step how each i4Q Solution works and what each of the configuration parameters means is also provided.

Document History

VERSION	ISSUE DATE	STAGE	DESCRIPTION	CONTRIBUTOR
0.1	04-Sept-2023	ToC	1st ToC draft	ITI
0.2	08-Feb-2024	Draft	Solutions overview	BIBA, BIESSE, CERTH, ENG, EXOS, FACTOR, FARPLAS, FIDIA, IBM, IKER, ITI, KBZ, RIASTONE, TUB, UNI, UPV, WHIRLPOOL
0.3	01-Mar-2024	Draft	Solutions requirements	BIBA, BIESSE, CERTH, ENG, EXOS, FACTOR, FARPLAS, FIDIA, IBM, IKER, ITI, KBZ, RIASTONE, TUB, UNI, UPV, WHIRLPOOL
0.4	10-Mar-2024	Draft	Solutions installation and configuration	BIBA, BIESSE, CERTH, ENG, EXOS, FACTOR, FARPLAS, FIDIA, IBM, IKER, ITI, KBZ, RIASTONE, TUB, UNI, UPV, WHIRLPOOL
0.5	05-Apr-2024	Draft	Solutions user manual	BIBA, BIESSE, CERTH, ENG, EXOS, FACTOR, FARPLAS, FIDIA, IBM, IKER, ITI, KBZ, RIASTONE, TUB, UNI, UPV, WHIRLPOOL
0.6	23-Apr-2024	Draft	Version ready for internal review	ITI
0.7	24-Apr-2024	Draft	Internal review	I-VLAB, FACTOR
0.8	26-Apr-2024	Draft	Draft version with modifications ready for final review	ITI
1.0	30-Apr-2024	Final Doc	Quality check, content review and issue of final document	CERTH

Disclaimer

Copyright message

TABLE OF CONTENTS

LIST OF FIGURES

ABBREVIATIONS/ACRONYMS

6LoWPAN	IPv6 over Low power Wireless Personal Area Networks
ACM	Advanced Cluster Management
AI	Artificial Intelligence
Amazon S3	Amazon Simple Storage Service
AP	Access Point
API	Application Programming Interface
CA	Certificate Authority
CPU	Central Processing Unit
CSV	Comma-Separated Values
DB	DataBase
DPPM	Defective Parts per Million
ETL	Extract, Transform and Load
FFT	Fast Fourier Transform
FMU	Functional Mock-up Units
GB	Gigabytes
GHz	Gigahertz
GUI	Graphical User Interface
HTTP	Hypertext Transfer Protocol
$i4Q^{AD}$	Analytics Dashboard
$i4Q^{AI}$	AI Models Distribution to the Edge
$i4Q^{BC}$	Blockchain Traceability of Data
$i4Q^{BDA}$	Big Data Analytics Suite
$i4Q^{DA}$	Services for Data Analytics
$i4Q^{DIT}$	Data Integration and Transformation Services
$i4Q^{DR}$	Data Repository
$i4Q^{DT}$	Digital Twin Simulation Services
$i4Q^{EW}$	Edge Workloads Placement and Deployment
$i4Q^{IM}$	Infrastructure Monitoring
$i4Q^{LRT}$	Manufacturing Line Reconfiguration Toolkit
$i4Q^{PA}$	Prescriptive Analysis Tools

i4Q^{PQ}	Data-Driven Continuous Process Qualification
i4Q^{QD}	Rapid Quality Diagnosis
i4Q^{QE}	QualiExplore for Data Quality Factor Knowledge
i4Q^{SH}	IIoT Security Handler
i4Q^{TN}	Trusted Networks with Wireless and Wired Industrial Interfaces
IEEE	Institute of Electrical and Electronics Engineers
IIoT	Industrial Internet of Things
IoT	Internet of Things
IP	Internet Protocol
IWSN	Industrial Wireless Sensor Network
JSON	JavaScript Object Notation
MB	Message Broker
mDNS	Multicast Domain Name System
ML	Machine Learning
OS	Operating System
PEM	Privacy-Enhanced Mail
RAM	Random Access Memory
REST	Representational State Transfer
RIDS	Reliable Industrial Data Services
SDN	Software Defined Networking
SQL	Structured Query Language
SSL	Secure Sockets Layer
TCP	Transmission Control Protocol
TLS	Transport Layer Security
TSH	Time-Slotted Channel Hopping
TSN	Time Sensitive Networking
UI	User Interface
UID	Unique Identifier
URL	Uniform Resource Locator
YAML	YAML Ain't Markup Language

Executive summary

i4Q Project aims to provide a complete set of solutions consisting of IoT-based Reliable Industrial Data Services (RIDS), the so called 22 i4Q Solutions (17 software tools and 5 guidelines), able to manage the huge amount of industrial data coming from cheap cost-effective, smart, and small size interconnected factory devices for supporting manufacturing online monitoring and control.

This deliverable provides a general analysis of the different i4Q Solutions, so that anyone without much knowledge of the project can understand what each of the solutions is and what actions can be carried out with each one of them.

The professional profiles are addressed for each one of the i4Q Solutions, so that companies know what type of personnel should be assigned to manage them.

The hardware and software requirements necessary to be able to deploy each one of the i4Q Solutions are indicated:

- Hardware requirements refer to the resources that the machine must have available to deploy a given solution, such as: free storage space, amount of RAM, number of CPUs and any other relevant information.
- Software requirements are those libraries and dependencies that need to be installed on a machine so that an i4Q Solution can be deployed and function correctly.

The installation and configuration of each one of the i4Q Solutions are also detailed, so that anyone can follow the steps and deploy them.

An explanation on how the user can obtain the source code or binary installation file of the corresponding i4Q Solution (either by cloning a GitLab repository or by downloading a set of artefacts compressed in a zip file) is provided.

A user's manual explaining step-by-step how each i4Q Solution works and what each of the configuration parameters means is also provided. The aim of this manual is to explain how each of the i4Q Solutions works.

Document structure

Section 1: Introduction. This section describes the purpose and approach of the Solutions Handbook in the i4Q Project.

Section 2: Solutions overview. In this section is provided an overview of each i4Q Solution, its main features and the professional profiles to which each of them is addressed.

Section 3: Solutions requirements. This section analyses the hardware and software requirements that the machine must fulfil in order to correctly install and deploy each of the i4Q Solutions.

Section 4: Solutions installation and configuration. In this section, the main steps for successfully install and configure each of the i4Q Solutions are briefly explained.

Section 5: Solutions user manual. This section is responsible for explaining the operation of the new functionalities implemented in the i4Q Solutions that are not included in the web documentation. For those solutions that have not been modified with respect to their web documentation, it is indicated where the user manual and other useful resources can be found.

Section 6: Conclusions. This section summarises the main results of the deliverable.

1. Introduction

The objective of this deliverable is to gather all the necessary information to be able to explain in a simple manner to a potential user the purpose of each one of the solutions of the i4Q Project. This project consists of 22 i4Q Solutions (17 software tools + 5 guidelines), however, this deliverable will be focused on the 17 software tools and the Message Broker communication tool. For each of these, it will be explained: what are the resources needed to make them work, what steps must be followed to install and properly configure each solution, and how each one of them works. In this way, it is intended that the user interested in an i4Q Solution will be able to deploy, install and configure it by following the steps indicated in the corresponding user manual without the need for prior knowledge of the i4Q Project. These brief explanations can be accompanied by references to the i4Q Project web documentation [1], which provides a more complete and detailed information on each of the i4Q Solutions.

First, there is a general description of each of the i4Q Solutions, their main characteristics, and the professional profile to which each of them is addressed, so that the user can understand whether a specific solution meets his/her needs and is aimed at someone with his/her professional profile. If not, the user may need to acquire a set of skills to be able to understand how the solution works and the results that it produces.

Afterwards, the hardware and software requirements necessary to deploy each of the i4Q Solutions on a machine are analysed. In this way, the user can know in advance what are the minimum resources that must have available on the machine to be able to deploy it correctly, as well as what dependencies or software libraries need to be installed before completing the installation of a solution.

Next, it is briefly explained from where the necessary artefacts can be obtained to install the solution on a user's local machine. In addition, a guide with the steps that must be followed to properly install and configure each of the i4Q Solutions is provided.

Finally, an updated user manual is provided in which the modifications that have been made to each of the i4Q Solutions since the last version of the user manual was published in the web documentation. These modifications will be later incorporated into the website. On the other hand, for those solutions that have not undergone any major change, the link to the official documentation of the solution and other links that may be of interest for understanding how it works are provided.

2. Solutions overview

This section aims to provide a general analysis of the different i4Q Solutions, so that anyone without much knowledge of the i4Q Project can understand what each of the solutions is and what actions can be carried out with each one of them. In addition to this, it is also indicated for which professional profiles is addressed each of the i4Q Solutions, so that companies know what type of personnel should be assigned to manage each one of them.

2.1 i4Q^{AD} – Analytics Dashboard

2.1.1 General description

The i4Q Analytics Dashboard (i4Q^{AD}) provides a graphical representation of extensive input data and information provided by other solutions. It relies on Apache Superset as the main visualisation technology since it covers the end user's desired visualisations as well as the technical requirements.

2.1.2 Main features

The main feature provided by i4Q^{AD} Solution is the capability to create customizable graphs in a customized i4Q theme dashboard. This solution is mapped to the Monitor and Diagnosis sub-component of the Platform Tier and to the Business Analytics for Quality of the Enterprise Tier, enabling the capability to visualise and analyse multiple data sources at the same time including input data and the information provided by the other solutions, as well as monitor the quality of the processes for the end-users.

2.1.3 Target audience

This solution is intended to be used by all the users, although users with a Data Analytics profile are the ones who can take more profit from this tool.

2.2 i4Q^{AI} – AI Models Distribution to the Edge

2.2.1 General description

The main overarching goal of the i4Q Project is to improve production processes towards a zero defects environment. That is achieved mainly through the introduction of IoT devices feeding real-time data to AI models. The i4Q^{AI} Solution is in charge of distributing the corresponding AI model to the edge where it becomes a component of the lifecycle management of edge component.

The edge components manage, deploy and monitor models at the edge. Then, the analysis components use the models to perform their inferences close to the data sources.

2.2.2 Main features

The i4Q^{AI} Solution supports managing the lifecycle of AI models, demonstrated via the Line Reconfiguration Tool (i4Q^{LRT}) Solution. Models get created in the cloud mostly by Data Scientists, and when ready they need to be deployed on the edge. i4Q^{AI} enables the full lifecycle management of AI models, from their creation in the cloud, through placement and deployment, while on the reverse path monitoring information flows to the cloud such that the efficiency of the current model can be assessed and if required a new version can be prepared and pushed to be deployed on the edge. Rolling updates are supported such that deployment of a new version of a model can be performed seamlessly without interference to the working system.

2.2.3 Target audience

This sub-section is intended for system administrators and integrators, in addition to Data Scientists. The intention is for each one to handle his area of expertise, while leaving the heavy lifting of the dynamic deployment to this component which operates at the back end without need for intervention from additional professionals such as developers.

2.3 i4Q^BC – Blockchain Traceability of Data

2.3.1 General description

i4Q^BC Solution provides tools to ensure data trustiness and full traceability, enhancing the level of trust by employing a blockchain-based data service, improving trust and acceptability by providing security and trust in the data that flows directly to the blockchain, serving as a single point of truth, preserving provenance, and supporting non-repudiation.

2.3.2 Main features

This solution adds a trusted layer on top of machine configurations that provides:

- **Provenance / Data lineage**. Recording and understanding all machine configuration changes the system underwent along the way: how, when, and by whom.
- **Authenticity / Non-Repudiation**. Proving that the machine configuration was received exactly as it was sent by the source, preventing authorship disputes.
- **Tamper- Evidence**. Detection of any changes to the protected recordings.
- **Multi-sig**. A mechanism to approve machine configuration only when signed by several designated parties (e.g., the operator and the manager).

2.3.3 Target audience

This solution is intended to be used by the following actors:

- **Operator**. A user who plans and identifies the machines and machine configurations that need to be managed with blockchain technology. According to the plan, the user can then initialise and request updates on machine configurations.
- **Controller**. A user who evaluates the update request on machine configurations. According to the evaluation, a change request can be approved or rejected.
- **Auditor**. A user who verifies and audits the history of machine configurations.

2.4 i4Q^BDA – Big Data Analytics Suite

2.4.1 General description

The i4Q^BDA Solution provides an interface for the creation of software bundles, which in turn are built and packaged in a way that simplifies the process of technology deployment on the user premises, without needing previous configuration, since these bundles come in a "plug-and-play" format. This solution provides a catalogue of software, ranging from databases, AI/ML technologies (such as Apache Spark), as well as the i4Q^DA Solution.

2.4.2 Main features

The main features of the i4Q^BDA Solution are the simple user interface, a catalogue of technologies that normally require some expertise to handle, configure and deploy, and the creation of technology bundles that remove some of the complexity that is normally associated with these types of technologies, especially AI/ML related technologies, such as Apache Spark. Since it also comes with the i4Q^DA Solution in its catalogue, this provides a simpler way of deploying this Solution on premises.

2.4.3 Target audience

This solution is intended to be used mainly by integration engineers, but it can be also used by any person with knowledge that is connected with these technologies such as Developers, Data Analysts or Data Scientists.

2.5 i4Q^DA – Services for Data Analytics

2.5.1 General description

The i4Q^DA Solution provides a simpler way to create AI/ML workflows, alongside their management and execution. By having an abstraction layer over the complexity associated with these types of technologies and the process of data analysis, this tool aims to increase the accessibility of AI to the normal user, without needing specific knowledge both in these technologies but also in programming. The data analytics environment provided by this tool allows the dynamic creation of data analytics workflows, in a seamless and intuitive manner. It also provides integration with already established Big Data and AI technologies, such as Apache Spark, TensorFlow or MLflow.

2.5.2 Main features

The main features of this solution are the focus on user support, by having a block based visual programming interface, allowing drag and drop functions for the workflow components, as a complexity abstraction layer, the fact that it is a no code solution, not requiring programming skills to use this tool, and also the fact that it's a standalone but extensible solution, since it comes already packed with all the necessary technologies and tools but it allows the integration of other code blocks that are user specific. Another feature of this solution is the fact that it is an open-source solution, built with only open-source technologies, providing a cheaper alternative to other similar solutions that are proprietary (and in consequence more expensive). This tool is also multitenant, since it is ready to be deployed in different environments, such as cloud and on premise.

Alongside the above features, this solution can also be characterised by the following components:

- **Workflow Creation Component**. Responsible for the creation, definition, and configuration of the Workflows, via User Interface (UI), where the blocks for the workflow tasks are provided.
- **Airflow Orchestration Component**. Integration of the $i4Q^{DA}$ Solution with Apache Airflow, the workflow orchestrator that will be in charge of managing and executing the workflows that are defined in the component above.
- **Operator Integration Interface**. Allows the integration of custom operators from the user, that can be used in the workflow creation and definition.
- **Technology Configuration Interface**. Enables the configuration of the integrated technologies, allowing the usage of external instances instead.

Finally, the blocks mentioned in the beginning of this section contain pre-built basic functionalities, that are necessary when performing data analysis. These functionalities are divided into three main groups, being the Connectors, blocks that allow data collection and ingestion from different data sources, the Transformers, providing all sorts of Extract, Transform and Load (ETL) functionalities, and the Analytics, which contain blocks for the training and serving of AI/ML models.

2.5.3 Target audience

The $i4Q^{DA}$ Solution currently is still focused on users that have some Data Analysis knowledge, but the continuous development will allow the usage of this tool by the general user.

2.6 $i4Q^{DIT}$ – Data Integration and Transformation Services

2.6.1 General description

The Data Integration and Transformation Services ($i4Q^{DIT}$) Solution performs the initial steps of an analytical pipeline. When working with sensor data, pre-processing is required to remove noise and deliver a dataset suitable for building predictive algorithms. $i4Q^{DIT}$ Solution is responsible for drawing data from machines, performing cleaning and correction actions and afterwards applying different pre-processing techniques and extracting features, that are then sent to other analytical solutions via the Message Broker.

2.6.2 Main features

The main features of the $i4Q^{DIT}$ Solution are summarized below. In general, the solution is user friendly, however it requires technical knowledge to apply some functions to choose the right preprocessing method.

- **GUI**. The solution shares a common UI with the other two solutions developed by CERTH ($i4Q^{IM}$ and $i4Q^{QD}$). Through this UI the user can load a dataset, apply a set of pre-processing functions suitable for this pilot and visualize graphs of filtered vs unfiltered signals.
- **Data import**. $i4Q^{DIT}$ Solution can import data from storage solutions ($i4Q^{DR}$) or directly from the machines.
- **Data cleaning and preparation**. This feature includes cleaning of wrong or duplicated values, computation of new variables, removal of outliers among others.

- **Pre-processing and feature extraction**. i4Q^DIT Solution has developed customized pre-processing techniques according to the pilot use case. Some indicative examples are Fast Fourier Transformation, Empirical Mode Decomposition, etc.

2.6.3 Target audience

The i4Q^DIT Solution delivers data suitable for further analysis by importing data, applying pre-processing and feature extraction, as well as data cleaning. The solution can be used by analysts and engineers that want to perform filtering and extract features from raw signals. For this reason, some technical knowledge is required to better comprehend the functionalities of the solution.

2.7 i4Q^DR – Data Repository

2.7.1 General description

The Data Repository (i4Q^DR) Solution offers a collection of toolkits to configure, bootstrap and manage a number of tools and technologies related to the storage and management of data, supporting different usage scenarios. This set of tools and technologies includes the following:

- An SQL relational database server, like MySQL, MariaDB, and PostgreSQL.
- A JSON document-oriented database server (MongoDB).
- A wide-column NoSQL distributed database server, appropriate for managing massive amounts of data (Cassandra).
- A graph database server that allows storing data relationships (Neo4J).
- An in-memory data structure store, used as a distributed, in-memory key-value database, cache, and Message Broker (Redis).
- A time-series SQL database (TimeScaleDB).
- A high-performance, Amazon S3-compatible object storage used to store general purpose files (MinIO).

2.7.2 Main features

i4Q^DR Solution offers a variety of configurations or scenarios of the supported data storage technologies. These scenarios can be seen as different ways of deploying the tools to meet some requirements related to aspects such as performance or security. More specifically, the following four scenarios have been defined:

- **Single server**. Offers the most basic version of the tool. It leverages a single instance of the tool, ready to be used. It is worth noting that such single instances do not consider any security issue beyond its default configuration. Thus, they are only recommended for the development tasks of a pilot or solution.
- **Single server with TLS security**. Offers the tool with a security configuration based on the use of TLS (with x509 certificates). This scenario is suitable for production settings in which a single instance of the tool suffices to provide a secure and stable service.
- **High availability**. Offers the tool in a high availability mode. This implies defining some cluster or replica set environment, composed by a number of instances of the tool that

work in a cooperative mode (for instance, a cluster of replicas of a database server). These instances do not offer any security mechanism beyond those that are offered by default and are recommended for development purposes only.

- **High availability with TLS security**. This is similar to the "High Availability" scenario, but it adds a security configuration based on the use of TLS. This is recommended in order to offer a secure, fault-tolerant, highly available service.

The i4Q^DR Solution relies substantially on the use of virtualisation techniques. For each one of the technologies and scenarios selected in a given setting, one or more Docker containers running an instance of the corresponding storage technology are built and launched.

This solution also involves the deployment of Trino, an open source, highly parallel and distributed SQL-compliant query engine that offers a relational-like view of different data storage tools. The purpose of this layer is dual: on one side, to improve the interoperability of the Data Repository with other i4Q Solutions and, on the other, to facilitate the support for other storage technologies in the future, if necessary. In addition, Trino allows the execution of federated queries, which means that several databases of different types (relational, object storage, streaming or NoSQL, etc.) can be accessed within the same query.

2.7.3 Target audience

Data Repository is a horizontal asset that offers its services to any other i4Q Solution that requires functionality related to the storage and management of any kind of data.

This solution can be used both manually and in an automated manner. First, any operator can easily interact with the tool to configure and launch the toolkits. Once the necessary scenarios have been set up, the operator can use and manage them via the interface that toolkits typically offer (such as command line consoles or web interfaces). In addition, the toolkits can also be used by other i4Q Solutions, via their usual connection channels, namely TCP/IP sockets or REST APIs. Furthermore, the solution offers an additional level of automated usage, which allows any i4Q Solution to interact with the configuration and deployment process of the toolkits, in order to customise, if necessary, any step of that process.

2.8 i4Q^DT – Digital Twin Simulation Services

2.8.1 General description

The i4Q^DT (Digital Twin Simulation Services) software allows to build different Digital Twin models of a manufacturing asset/plant, visualizing its results in a user-friendly environment in different formats (graphs/tables).

The i4Q^DT is comprised of three main software packages: physics-based back end, data-driven back end, and user interface front-end. The physics-based workflow makes use of Functional Mock-up Units (FMU) that have been compiled from different modelling languages like Modelica, which are component-oriented and based on a set of equations defining the physics behavior of the system. The data-driven approach corresponds to machine learning methods, comprising data-driven machine learning techniques, which are highly promising since a model learns critical insights directly and automatically from the given datasets.

2.8.2 Main features

i4Q^{DT} is a user-friendly Digital Twin model building tool that provides the user with the following functionalities:

- **Building models**. The tool allows the user to build different models, both physics-based ones through pre-existing FMU models and data-driven ones through machine learning models. In the case of the physics-based models, it allows also to establish the relationships between the inputs of the model and the collected data (contextualization). This way, starting from a set of previously created FMU independent models, the user can build a more complex model.
- **Model storing and loading**. Provides the capability of loading, storing and updating individual models representing the different sections and machines of a plant.
- **Model simulation**. Provides the capability of running simulations of both data-driven models (machine learning Python models) and physics-based models (FMU compiled models.
- **Result visualization**. The tool allows the user to visualize its results in a user-friendly environment in different formats (graphs/tables).
- **GUI**. The user-friendly guides a non-expert in the process of building/training the model, both in its physics-based workflow and its data-driven workflow. All the design and parameterization of the model is done step by step. The GUI also allows the user to visualize the results when the model is simulated.

2.8.3 Target audience

The i4Q^{DT} Solution, through its user interface, is suitable for any person from a design department in a manufacturing environment, including engineers, product responsible and data analysts. Some technical knowledge regarding model building (both physics-based and data-driven) is required from the user. In the physics-based workflow, it is also required to have previously created models in FMU format, to use them to create a more complex one.

The user-friendliness and configurability of the solution allows for an easier and to-the-point building of the models, the user being able to obtain quick results that could guide the decision-making regarding different manufacturing processes.

2.9 i4Q^{EW} – Edge Workloads Placement and Deployment

2.9.1 General description

The main overarching goal of the i4Q Project is to improve production processes towards a zero defects environment. That is achieved mainly through the introduction of IoT devices feeding real-time data to AI models operating via APIs to driver workloads. i4Q^{EW} is in charge of distributing the corresponding AI workloads to the edge where it becomes a part of the lifecycle management of edge components. AI workloads and models have different lifecycle thus different components handle ease such capability.

The edge components manage, deploy, monitor, AI workloads on the edge. Models and workloads are later used by the analysis components to run their inference close to the data sources.

Whenever needed AI workloads can be updated in a rolling manner such that new versions get deployed while still serving existing applications.

2.9.2 Main features

i4Q^EW Solution supports managing the lifecycle of AI workloads, demonstrated via the i4Q^LRT Solution. Workloads get created in the cloud mostly by developers, and when ready they need to be deployed on the edge. i4Q^EW enables the full lifecycle management of AI workloads, from their creation in the cloud, through placement and deployment, workloads enable easy access to underlying models via agreed upon APIs. Rolling updates are supported such that deployment of a new version of an AI workload can be performed seamlessly without interference to the working system.

2.9.3 Target audience

This sub-section is intended for system administrators and integrators, in addition to data scientists. The intention is for each one to handle his area of expertise, while leaving the heavy lifting of the dynamic deployment to this component which operates at the back-end without need for intervention from additional professionals such as developers. Interaction may be required with the i4Q^AI to ensure that the AI workloads and models get to collaborate smoothly.

2.10 i4Q^IM – Infrastructure Monitoring

2.10.1 General description

The Infrastructure Monitoring (i4Q^IM) Solution utilizes machine learning algorithms to effectively correlate the presence of machine malfunctions with the real-time data deriving from machine sensors. Upon detecting a problematic machine state, the solution provides a notification, informing the operator to take corrective actions to avoid devastating machine breakages.

2.10.2 Main features

The i4Q^IM Solution strives to assist the machine operators in their tasks in an intuitive way, ensuring operational efficiency and ease of interaction. The following points summarize the core functionalities provided by the i4Q^IM Solution.

- **GUI**. The i4Q^IM Solution offers its key features in a user-friendly interface, allowing user interaction and providing insightful visualizations to present the results of the performed analysis. The UI shares a common structure with the i4Q^DIT and i4Q^QD, to maintain consistency across all CERTH's solutions.
- **Data Import**. The user has the option to upload data from several different sources, for static or dynamic data analysis. The user can load CSV data files from local directories, from collections through the MongoDB provided by i4Q^DR, or from the Message Broker.
- **Static data analysis**. A user that is familiar with machine learning operations can split the loaded data into train/test sets to train a new predictive model or select a pretrained one to evaluate the presence of machine malfunctions.
- **Dynamic data analysis**. A less experienced user with machine learning operations, can use the dedicated operator's tab to view the analytic results of a pre-trained ML model, that

evaluates the machine status in real-time. The results are also automatically stored in the i4Q^{DR} for further study.

- **Notifications & Visualizations**. Upon the identification of a machine problem, the tool provides an alert, indicating the percentage of samples in which a problem has been detected. Moreover, information about the model's inner workings is also provided as feature importances. The importance indicates the magnitude of contribution that its feature (sensor) has towards the prediction of the model.

2.10.3 Target audience

i4Q^{IM} can serve the needs of infrastructure personnel regardless of their expertise in machine learning. A data analyst or an engineer can use the static data analysis functionality of the tool, to explore past machine malfunction cases by studying historical data. Also, they can benefit from the option of training new ML models, since the tool can be integrated and adjusted to operate under different machines. On the other hand, the operators can see the live analytics, to have a continuous overview of the machine status and be alerted of any potential failures during their shift.

2.11 i4Q^{LRT} – Manufacturing Line Reconfiguration Toolkit

2.11.1 General description

This solution is a toolkit that allows the use of algorithms as services from different data clients. The aim of these algorithms is to provide end users with different solutions to help in the reconfiguration of the line.

2.11.2 Main features

The i4Q^{LRT} Solution enhances end users' ability to reconfigure lines efficiently through the implementation of various heuristic algorithms. The following points elaborate on the key functionalities offered by the solution:

- **Intuitive Graphic User Interface** (GUI). i4Q^{LRT} presents a user-friendly interface, facilitating direct interaction with the algorithm and real-time data visualization. This GUI shares a consistent structure with other i4Q Solutions such as i4Q^{DT} and i4Q^{PA}, ensuring coherence across the suite.
- **Versatile Data Import**. Users can seamlessly import data from diverse sources, supporting both static and dynamic data analysis. Whether it's CSV files from local directories or data streams from the Message Broker, the solution accommodates various data formats.
- **Manual Algorithm Execution**. Users have the flexibility to manually execute the algorithm, allowing for testing and evaluation of different scenarios to gauge performance.
- **Real-time Dynamic Data Analysis**. Through the dedicated operator tab, users can access analytical insights generated by the algorithm in real time. This feature leverages data directly from the Message Broker (Kafka), enabling swift decision-making based on up-to-the-minute information.
- **Historical Data Access**. The solution offers a comprehensive history feature, enabling users to retrieve and review past data for retrospective analysis and reference.

By encompassing these functionalities, the i4Q^LRT Solution empowers users with robust tools for line reconfiguration and optimization, enhancing operational efficiency and decision-making capabilities.

2.11.3 Target audience

The target audience of the i4Q^LRT Solution is composed of production managers, line engineers, data analysts and plant operators in manufacturing companies. These professionals seek to optimise the efficiency and performance of production lines by implementing advanced heuristic algorithms. They have a medium to high level of experience in the use of data analysis tools and are familiar with the basic concepts of industrial process optimisation. Their main objective is to reduce downtime, improve resource utilisation and increase productivity in your manufacturing operations.

2.12 i4Q^PA – Prescriptive Analysis Tools

2.12.1 General description

The Prescriptive Analysis Tool (i4Q^PA) provides mainly simulations as a service. It exploits the Digital Twin developed to test different configuration parameters and analyse the effect small changes can generate in production. It makes use of manufacturing resources, production planning and process condition. The prescriptive analysis will come from exhaustive simulation with no specific optimization algorithm implemented.

2.12.2 Main features

i4Q^PA is a user-friendly prescription tool that provides the user with the following functionalities.

- **GUI**. The user-friendly interface contains all the information required for non-experts to easily prescribe the optimum configuration of a model, or to visualize the behaviour of a system under different conditions.
- **Exhaustive simulations**. Once the user has selected the parameters to be varied and their range of values, the solution carries out one simulation for each parameters' values combination, which results in a significant number of simulations.
- **Simulations evaluations**. The user can evaluate all the carried-out simulations with a predefined or custom evaluation function.
- **Model prescription**. The user is presented with the optimum prescription of the model, along with the best alternatives for the optimum. In addition, the user will also be able to check the worst configurations of the system in the same tab.

2.12.3 Target audience

The user-friendliness and configurability of the solutions makes it suitable for any manufacturing personnel, starting with machine operators, to engineers, managers, or data analysts. However, operators or managers will require for the machine/system to be previously modelled.

2.13 i4Q^{PQ} – Data-Driven Continuous Process Qualification

2.13.1 General description

The i4Q^{PQ} (Data-Driven Process Qualification) Solution improves structural quality control in manufacturing by harnessing the power of real-time data analysis. Designed to be intuitive, it allows even those with minimal technical expertise to monitor and ensure their production processes meet strict quality standards. By connecting directly to data sources and selecting key variables, users can instantly verify if their operations are producing within desired quality ranges. This innovative tool is essential for businesses aiming to maintain excellence in product reliability and consistency, making it a vital asset for industries where precision is paramount, such as aerospace, automotive, and medical device manufacturing.

2.13.2 Main features

i4Q^{PQ} is a user-friendly forecasting and real-time monitoring tool to describe the current behaviour of single quality measurements of manufactured products. It provides automated multiple features, described as follows:

- **GUI**. Provides guidance for the novel approach to forecast the process qualification and acts as main integration part for other software. Furthermore, it is also extendable if other software is demanded.
- **Data Connection**. Provides an interface to popular storage systems like MongoDB, HTTP or CSV files as well as real-time connection to Kafka.
- **Visualizations**. Provides two established visualizations (Process Chart and Histogram) to find observations outside the quality range and interpret statistical measurements across the complete range of the scaled distribution.
- **Distribution interpretation**. Provides information about the current distribution and the number of Defective Parts per Million (DPPM).
- **Forecast**. Provides AutoML capabilities and individual adjustments to forecast the Process Qualification and calculate the C_{pk} value.

2.13.3 Target audience

The i4Q^{PQ} Solution is tailored for professionals within the manufacturing sector who are directly involved in the oversight of production quality and process efficiency. This includes Quality Control Managers, Process Engineers, and Production Supervisors who seek to leverage real-time data to ensure that manufacturing operations adhere to the highest quality standards. Additionally, Data Analysts and IT Specialists in the manufacturing industry will find i4Q^{PQ} 's data integration and analysis capabilities invaluable for optimizing process performance and output quality. By providing a user-friendly interface and powerful analytical tools, i4Q^{PQ} is designed to empower these professionals with the insights needed to make informed decisions, improve process reliability, and maintain competitive advantage through superior product quality.

2.14 i4Q^{QD} – Rapid Quality Diagnosis

2.14.1 General description

The Rapid Quality Diagnosis (i4Q^{QD}) Solution aims to assist in quality control processes through the exploitation of machine learning algorithms. The service allows for the identification of non-conforming products or sub-optimal manufacturing conditions by analysing the signals coming from the machine sensors. This is feasible, via ML predictive models that are trained on historical sensor data, effectively capturing their correlation to the occurrence of past manufacturing issues.

2.14.2 Main features

The primary aim of the i4Q^{QD} Solution is to support machine operators in their tasks through an intuitive interface, promoting operational efficiency and smooth interaction. The following points outline the fundamental features offered by the i4Q^{QD} Solution:

- **GUI**. The i4Q^{QD} Solution presents its key functionalities through a user-friendly interface, facilitating user interaction and offering insightful visualizations to communicate analysis results effectively. The UI maintains a consistent structure with other CERTH solutions, such as i4Q^{DIT} and i4Q^{IM}, ensuring coherence across all platforms.
- **Data Import**. Users can import data from various sources for both static and dynamic data analysis. This includes loading CSV data files from local directories, accessing collections through the MongoDB provided by i4Q^{DR}, or retrieving data from the Message Broker.
- **Static Data Analysis**. For users proficient in machine learning operations, the solution allows the splitting of loaded data into train/test sets for training new predictive models or selecting pre-trained models to detect product defects or production issues.
- **Dynamic Data Analysis**. Users less familiar with machine learning operations can utilize the dedicated operator's tab to observe real-time analytic results of pre-trained ML models that assess the quality status of the production process. These results are automatically stored and accessible through the i4Q^{DR} for further examination.
- **Notifications & Visualizations**. In case of production issues or product defects, the tool provides alerts indicating the percentage of samples where a problem has been detected. Additionally, it provides information regarding the model's internal mechanisms, including feature importances that signify the degree of contribution each feature (sensor) has towards the model's predictions.

2.14.3 Target audience

The i4Q^{QD} Solution is designed to meet the needs of infrastructure personnel, regardless of their level of expertise in machine learning. Whether it's a data analyst or an engineer, individuals can utilize the tool's static data analysis functionality to delve into past instances of product defects by examining historical data. Furthermore, they can capitalize on the option to train new ML models, as the tool can be seamlessly integrated and adjusted to operate across different machines. On the other hand, operators can rely on live analytics to maintain a continuous overview of machine status and receive timely alerts about any potential production process issues during their shift.

2.15 i4Q^{QE} – QualiExplore for Data Quality Factor Knowledge

2.15.1 General description

The i4Q^{QE} Solution is a web-based software tool for visualising information quality characteristics and factors. Producers need to know the latter to assess their relevance for production and identify measures to manage them. i4Q^{QE} is an example of a measure to raise awareness of data quality. The Deliverable D3.9 [2] outlines these and other measures in its proposed activity framework. QualiExplore's main application is standalone using a permissive open-source license (Apache 2.0), so stakeholders can easily exploit it. A proprietary chatbot complements the solution to simplify the access to the i4Q^{QE} knowledge base.

2.15.2 Main features

i4Q^{QE} has a *2-staged user interface* to grant users access to factor knowledge. The first stage serves as a filter because a high number of factors can cause information overload for users. Relevant *filter categories* include the user's goals, quality (information characteristics), and channels/sources. The goals include the perspective of the information user and the information creator/author. This approach is helpful because it emphasizes that many measures that avoid quality problems require both parties' involvement. Each category has several statements representing the user's interest in information quality problems and related factors. The indicated *factor categories* structure the factors and provide a link between statements and factors. Users can ask a chatbot about information quality instead of using the main graphical interface. This kind of interaction is more intuitive and, in some situations, faster. If users require a quick overview or if they are experienced users, the main graphical interface is likely faster.

2.15.3 Target audience

The main user groups of i4Q^{QE} are employees that author/create information and those that design data processing tools. The former gain awareness for information quality issues and may act accordingly to avoid them in the future. The latter benefit from an increased awareness because they may better anticipate potential information quality issues and counter them with technical or methodological measures in their data processing tools.

2.16 i4Q^{SH} – IIoT Security Handler

2.16.1 General description

i4Q^{SH} is a security solution that serves as a robust shield safeguarding the integrity and confidentiality of system communications. i4Q^{SH} assumes the pivotal role of generating Certificate Authorities (CAs) tailored to the specifications provided by the system's pilots. These customised CAs are produced to meet security standards, ensuring a fortified defence against potential threats.

2.16.2 Main features

The main feature of the i4Q^{SH} Solution is the generation of a CA for each pilot. This feature allows pilots to secure their communications effectively, as the other components of the solution leverage these individualized CAs to generate the certificates necessary for different functions.

By provisioning specific CA for each pilot, i4Q[SH] ensures that the solution aligns with the specific needs and requirements of each user. The main advantages of using these CAs are:

- Robust authentication mechanisms.
- Enabling secure access control.
- Identity verification.
- Encryption.
- Enhance the integrity and confidentiality of communications.

2.16.3 Target audience

This solution is aimed at developers, as it provides the Certificate Authorities (CAs) necessary to generate the required certificates for the solutions. Using these CAs they can establish secure connections, authenticate users, and encrypt data transmission effectively within their applications or systems. They can also implement robust security measures, ensuring that sensitive information remains confidential and protected against unauthorized access.

2.17 i4Q[TN] – Trusted Networks with Wireless and Wired Industrial Interfaces

2.17.1 General description

The i4Q[TN] Solution is a software-defined industrial interface for data communication that provides data exchange characterized by predictability and determinism, low latency, high reliability, trustability and low consumption.

2.17.2 Main features

The solution is focused on both wireless and wired interface that ensures high-quality data collection, providing connectivity to industrial data sources through Trusted Networks. The wireless part of the solutions is based on a Software-Defined Wireless Sensor Network that creates an adaptative mesh topology to provide connectivity to industrial sensor and interfaces, while the wired part provides connectivity to time-sensitive applications using Time Sensitive Networking (TSN) that is capable of identifying the needed quality of service configuration settings to achieve application specific latency and jitter requirements for the network layer.

2.17.3 Target audience

The main user groups for the Trusted Network solution are solution or system providers that require network connectivity with specific and restrictive requirements. Depending on the final use case, the user may require a determinist wireless network to provide connectivity to isolated industrial processes or sensors or to connect a closed loop time sensitive application over Ethernet.

2.18 Message Broker

2.18.1 General description

The Message Broker acts as an intermediary between the different solutions provided in the i4Q suite, as it provides a communication interface to streamline the exchange of real-time data. The Message Broker is based on Apache Kafka, which allows reliable, scalable, and real-time streaming of data.

2.18.2 Main features

The key features provided by the Message Broker are summarized below:

- **Kafka Broker**. This plays a central role, serving as a distributed system responsible for storing and managing topics, which are partitions of data streams. Additionally, it handles client requests from data producers and consumers, while ensuring fault tolerance through data replication.

- **Kafka Clients**. These interfaces provide a means for applications to interact with the Kafka server, enabling the publishing and consuming of messages. Producers use client interfaces to publish messages to topics, while consumers use them to subscribe to topics and retrieve messages for processing. The Message Broker provides both a Kafka Python client as well as a Kafka Flask API client.

2.18.3 Target audience

The Message Broker is not intended to be used as standalone tool by any of the infrastructure personnel, since as an online data streaming platform, its only purpose is to serve the communication needs of the i4Q Solutions for real-time data exchange.

3. Solutions requirements

The objective of this section is to know the **hardware** and **software requirements** necessary to be able to deploy each of the i4Q Solutions. **Hardware requirements** refer to the resources that the machine must have available to deploy a given solution, such as: free storage space, amount of RAM, number of CPUs and any other relevant information. **Software requirements** are those libraries and dependencies that need to be installed on a machine so that an i4Q Solution can be deployed and function correctly.

3.1 i4Q^{AD} – Analytics Dashboard

For the correct deployment of the i4Q^{AD} Solution, a machine with at least the following hardware requirements characteristics is needed: a CPU with 4 cores, 16 GB of RAM, and hardware virtualisation.

In terms of software requirements, the machine must have installed a Linux operating system capable of running scripts through Bash, the Git library and both Docker and the Docker Compose plugin.

3.2 i4Q^{AI} – AI Models Distribution to the Edge

For the correct deployment of the i4Q^{AI} Solution, the machine needs to fulfil the following hardware requirements: a CPU with 8 cores, 32 GB of RAM, and at least 10 GB of free disk storage. It is worth mentioning that these are the minimum requirements that the machine must satisfy for the solution to have an acceptable performance. However, the greater the quantity of resources provisioned by the machine, the better will be the performance of this solution.

Regarding the software requirements, the machine must have installed a Linux operating system, the Advanced Cluster Management (ACM) software, as well as the Kubernetes platform for managing containerised workloads and services.

3.3 i4Q^{BC} – Blockchain Traceability of Data

In order to be able to deploy the i4Q^{BC} Solution without problems, the machine must have a minimum of 16 GB of free disk storage, 16 GB of RAM and an Intel® Core™ i5-6500. It is recommended to have a set of servers capable of running a database cluster. For production at least 3 are required for fault tolerance, however 5 servers are recommended.

As for the software dependencies, the machine needs to have Linux system and installed the Git library and both Docker and the Docker Compose plugin.

3.4 i4Q^{BDA} – Big Data Analytics Suite

In terms of hardware requirements, the machine where the i4Q^{BDA} Solution is to be deployed needs to have a minimum of 8 GB of RAM, and at least an Intel® Core™ i5-6500 or an AMD Ryzen 5 3600 processor.

As for the software dependencies, the machine needs to have installed the Git library and both Docker and the Docker Compose plugin.

3.5 i4Q^DA – Services for Data Analytics

In order to be able to deploy the i4Q^DA Solution without problems, the machine must have a minimum of 16 GB of free disk storage, 16 GB of RAM and an Intel® Core™ i5-6500 or an AMD Ryzen 5 3600 processor.

For downloading the dockerised image of the solution from the GitLab repository to the machine, the Git library must have been previously installed and, given that a container must be created from the downloaded image, both Docker and the Docker Compose plugin must also be installed on the machine.

3.6 i4Q^DIT – Data Integration and Transformation Services

Regarding hardware requirements, to be able to deploy the i4Q^DIT Solution smoothly, it is recommended that the machine has a minimum of 4 GB of RAM, and a minimum of 10 GB of free storage, as the Docker image occupies a minimum of 5 GB. There is no minimum processor or number of cores required to run the solution. However, the more cores the machine has, the better performance the solution will have during its execution, so it is recommended to have a minimum of 2 cores.

In terms of software requirements, the machine needs to have installed the Git library (necessary to download the Docker image from the corresponding GitLab repository), and both Docker and the Docker Compose plugin to generate the container from the downloaded image.

3.7 i4Q^DR – Data Repository

In order to be able to deploy the i4Q^DR Solution without problems, it is necessary that the machine meets some minimum requirements. In terms of RAM, at least 4 GB is required, but the performance of this solution will improve as the number of gigabytes increases. As for the amount of free storage, there is no minimum number of gigabytes that the machine must have but, as it is a data storage solution, it is assumed that the available storage space must be larger than in the case of the other solutions, so it would be advisable to have at least 50 GB of free space on the machine. In terms of processing power, there is no minimum version or number of cores that the machine must have but given that data reading and writing operations usually require a certain processing capacity, it is estimated that the machine should have at least 2 cores. In this case, the higher the number of cores, the better will be the performance of the solution.

In terms of software requirements, the machine on which the solution is going to be deployed will need to have the following installed:

- **Bash** command line interface, which will allow the execution of scripts.
- **Docker Engine** and the **Docker Compose** plugin, which will be responsible for deploying and running the containers. In this case, any recent version compatible with YAML files from Docker Compose v3.9 will be enough.
- **OpenSSL** library. This will be responsible for creating the SSL test artifacts (keys, certificates, etc.).

- **Curl** library. It will allow to obtain the necessary files to build the Docker images.
- **Jq** library. It will be in charge of handling JSON files. It is required by the toolkits for MinIO and MongoDB.

3.8 i4Q^{DT} – Digital Twin Simulation Services

For a smooth deployment of the i4Q^{DT} Solution, a set of minimum hardware requirements must be met. In this case, it is estimated that a minimum of 4 GB of RAM is necessary (requirement for installing Docker), and a minimum of 20 GB of free storage, as the Docker image occupies around 10 GB. There is no minimum recommended processor or number of cores to run the solution. However, the more cores the machine has, the better will be the performance of the solution during its execution. Therefore, it is recommended that the machine has at least 2 cores.

For downloading the dockerised image of the solution from the GitLab repository to the machine, the Git library must have been previously installed and, given that a container must be created from the downloaded image, both Docker and the Docker Compose plugin must also be installed on the machine.

3.9 i4Q^{EW} – Edge Workloads Placement and Deployment

For the successful deployment of the i4Q^{EW} Solution, the machine needs to satisfy a set of hardware requirements: a CPU with 8 cores, 32 GB of RAM, and at least 10 GB of free disk storage. It is important to mention that these are minimum requirements that the machine must satisfy for the solution to have an acceptable performance. However, the larger the resources provided by the machine, the better will be the performance of this solution.

Regarding the software requirements, the machine must have installed a Linux operating system, the Advanced Cluster Management (ACM) software, as well as the Kubernetes platform for managing containerised workloads and services.

3.10 i4Q^{IM} – Infrastructure Monitoring

Regarding hardware requirements, to be able to deploy the i4Q^{IM} Solution smoothly, it is recommended that the machine has a minimum of 4 GB of RAM, and a minimum of 10 GB of free storage, as the Docker image occupies a minimum of 5 GB. There is no minimum processor or number of cores required to run the solution. However, the more cores the machine has, the better performance the solution will have during its execution, so it is recommended to have a minimum of 2 cores.

In terms of software requirements, the machine needs to have installed the Git library (necessary to download the Docker image from the corresponding GitLab repository), and both Docker and the Docker Compose plugin to generate the container from the downloaded image.

3.11 i4Q^{LRT} – Manufacturing Line Reconfiguration Toolkit

In the case of the i4Q^{LRT} Solution, a set of minimum and recommended hardware requirements are listed. Regarding the minimum requirements, it is indicated that the machine where the solution is going to be deployed must have at least 4 GB of RAM, 20 GB of available disk storage, and at least a CPU with 2 cores with a frequency of 2 GHz per core. In terms of recommended requirements, it is stated that the machine should have 8 GB of RAM, 50 GB of available disk storage, and a CPU with 4 cores with a frequency of 2.5 GHz per core.

In terms of software requirements, the machine needs to have installed the Git library (necessary to download the Docker image from the corresponding GitLab repository), and both Docker and the Docker Compose plugin to generate the container from the downloaded image.

3.12 i4Q^{PA} – Prescriptive Analysis Tools

For a smooth deployment of the i4Q^{PA} Solution, a set of minimum hardware requirements must be met. In this case, it is estimated that a minimum of 4 GB of RAM is necessary (requirement for installing Docker), and a minimum of 10 GB of free storage, as the Docker image occupies around 3 GB. There is no minimum recommended processor or number of cores to run the solution. However, the more cores the machine has, the better will be the performance of the solution during its execution. Therefore, it is recommended that the machine has at least 2 cores.

For downloading the dockerised image of the solution from the GitLab repository to the machine, the Git library must have been previously installed and, given that a container must be created from the downloaded image, both Docker and the Docker Compose plugin must also be installed on the machine.

3.13 i4Q^{PQ} – Data-Driven Continuous Process Qualification

A sufficient installation and deployment of the i4Q^{PQ} Solution requires a minimum hardware. Since the data connection and calculation in the solution depends on the inserted and demanded parameters, only an estimation can be derived for further usage. Thus, a minimum of 4 GB of RAM is necessary and a minimum of 10 GB of free storage. Regarding other hardware, there is no specific requirement. Nonetheless, more computing power leads to better performance of the solution.

To obtain the dockerised image of the solution from the GitLab repository on the machine, it is required that the Git library is already installed. Furthermore, in order to generate a container from the acquired image, the machine must also have Docker and the Docker Compose plugin installed.

3.14 i4Q^{QD} – Rapid Quality Diagnosis

Regarding hardware requirements, to be able to deploy the i4Q^{QD} Solution smoothly, it is recommended that the machine has a minimum of 4 GB of RAM, and a minimum of 10 GB of free storage, as the Docker image occupies a minimum of 5 GB. There is no minimum processor or number of cores required to run the solution. However, the more cores the machine has, the better performance the solution will have during its execution, so it is recommended to have a minimum of 2 cores.

In terms of software requirements, the machine needs to have installed the Git library (necessary to download the Docker image from the corresponding GitLab repository), and both Docker and the Docker Compose plugin to generate the container from the downloaded image.

3.15 i4Q^QE – QualiExplore for Data Quality Factor Knowledge

In order to correctly deploy the i4Q^QE Solution it is recommended that the machine has a minimum of 4 GB or RAM (minimum requirement to run Docker). There is no specified a minimum amount of free storage that the machine must have, but a space of at least 10 GB would be recommended to be able to download the Docker image and generate the container from it. The same goes for the processor, there is no recommended version or minimum number of cores. However, the more cores the machine has, the smoother the performance of the solution will be during its execution. Therefore, it is recommended that the machine has at least 2 cores.

As for the software dependencies, the machine needs to have previously installed the Git library and both Docker and the Docker Compose plugin.

3.16 i4Q^SH – IIoT Security Handler

For the correct functioning of this solution, no minimum requirements have been established in terms of hardware. However, the machine should have a minimum of 10 GB of free storage, as the installation of the solution and its corresponding software dependencies will require a certain amount of free disk space. In addition, it is intended that this solution can work in a smooth way, so it is estimated that at least 4 GB of RAM and a CPU with at least 2 cores should be provided to achieve this purpose.

Regarding the software requirements, the machine where the i4Q^SH Solution is going to be deployed must have installed at least the Git library to download the solution from the corresponding GitLab repository, and the OpenSSL library to allow the correct functioning of the solution.

3.17 i4Q^TN – Trusted Networks with Wireless and Wired Industrial Interfaces

The deployment of the i4Q^TN Solution is a little bit different from the rest of the i4Q Solutions, as it is based on two main isolated components: the **wireless** and **wired industrial interfaces**. Each of these components is oriented to different use cases and has a different set of system requirements.

3.17.1 System requirements for Wireless interfaces

1. **IWSN Node**: hardware platform based on CC2538 TI transceiver for each node to be deployed.
2. **IWSN Gateway**: SBC with IEEE 802.15.4 interface (i.e Raspberry Pi 4 with CC2538 transceiver using a tunslip6).

3. Different industrial or IoT sensors to be connected to the HUB platform (4-20mA, 0-10V, 24V DI, Modbus/RTU, i2C).

3.17.2 System requirements for Wired interfaces

1. **TSN Nodes**: hardware platforms (switches, bridges, end systems) supporting collection of standards from IEEE 802.1 TSN (e.g., IEEE 802.1Qbv, IEEE 802.1Qcc, IEEE 802.1AS).
2. Data transfer via Ethernet.
3. Windows 10 or Linux Ubuntu 18.04 for running the offline TSN Network Configuration Tool.

3.18 Message Broker

Regarding hardware requirements, to be able to deploy the Message Broker in a smooth way, it is recommended that the machine has a minimum of 4 GB of RAM, and a minimum of 20 GB of free storage, as the Docker image occupies a minimum of 8 GB. There is no minimum processor or number of cores required to run the solution. However, the more cores the machine has, the better performance the solution will have during its execution, so it is recommended to have a minimum of 2 cores.

In terms of software requirements, the machine needs to have installed the Git library (necessary to download the Docker image from the corresponding GitLab repository), Docker and the Docker Compose plugin, as well as the YAML and JSON processors (yq and jq, respectively).

4. Solutions installation and configuration

The idea of this section is to generate a step-by-step guide on how to install and configure each one of the i4Q Solutions, so that the user who wants to employ any of them can follow the steps and deploy them.

First of all, an explanation on how the user can obtain the source code or binary installation file of the corresponding i4Q Solution (either by cloning a GitLab repository or by downloading a set of artefacts compressed in a zip file) is provided.

Next, the installation of the i4Q Solution is explained (either by deploying a Docker container or by executing a set of scripts).

Finally, the configuration required to run the solution is indicated. In case the solution is designed to work in different scenarios or using different configuration methods, these can also be explained.

4.1 i4Q^{AD} – Analytics Dashboard

For installing the i4Q^{AD} Solution, it is necessary to access its GitLab repository [3] and download the zip file or clone the content on the machine where it will be installed.

Once the download is complete, the solution configuration process can start. For this, two different configuration procedures can be followed.

- **Running only the Apache Superset dashboard**. For this procedure, a prebuilt docker image from the i4Q Project's GitLab registry can be used. The steps to follow in this case are:

1. Open a command terminal.
2. Login into the GitLab registry using the command:

   docker login registry.gitlab.com

3. Pull the i4qdasboard Docker image using the command:

   docker pull registry.gitlab.com/i4q/ad/i4qdashboard:latest

4. Generate a container from the downloaded Docker image by using the command:

   docker run -d -p 8080:8088 --name i4qdashboard \

   registry.gitlab.com/i4q/ad/i4qdashboard:latest

5. To access the solution interface, open a web browser and enter the URL: `http://localhost:8088.`

- **Running Apache Superset with Apache Kafka and Apache Druid services**. For this procedure, follow these steps:

1. Open a command terminal.
2. Login into the GitLab registry using the command:

   docker login registry.gitlab.com

3. Run the Docker compose file with the following command:

   docker-compose -f docker-druid-kafka.yml up –build -d

4. After executing the above command, Apache Superset will be deployed on port 8088 and Apache Druid service will be deployed on port 8888.

5. For accessing the web interface of any of these services, open a web browser and enter the URL `http://localhost:8088` to visualise the Apache Superset interface or the URL `http://localhost:8888` to display the Apache Druid interface.

4.2 i4Q^{AI} – AI Models Distribution to the Edge

Note: The step-by-step user manual can be found in the Gitlab repository [4]. For a combined demo with the i4Q^{LRT} Solution, please refer to the i4Q^{LRT} Solution GitLab repository [5]. Requirements include creation, import, attachment and controlling deployment locations via label-based policies.

Note: Main underlying technology used by this solution is ACM (Advanced Cluster Management), which is in charge of the heavy lifting of artefacts. Artefacts accepted includes Kubernetes-based structures.

4.3 i4Q^{BC} – Blockchain Traceability of Data

Note: The Docker images for the i4Q^{BC} Solution can be found in the Docker Hub [6] as well as in the container registry of the i4Q Project's GitLab repository [7]. The i4Q^{BC} Solution Docker Compose file can be downloaded from the solution's GitLab repository [8].

Once it has been described from where the different artefacts can be obtained, the installation of the solution can be done by following these steps:

1. Deploy the Hyperledger Orion server following the Deployment Instructions specified in the corresponding page of the Hyperledger Orion Documentation [9].

2. Create Crypto Materials for users of Hyperledger Orion server following the guidelines specified in the corresponding page of the Hyperledger Orion Documentation [10].

3. Navigate to the location where the `docker-compose.yml` file is located and execute the command:

```
docker compose up -d
```

4. Open a web browser and enter the URL `http://localhost:8714/`.

For more technical details check the GitLab repository of the solution.

4.4 i4Q^{BDA} – Big Data Analytics Suite

To install the i4Q^{BDA} Solution, it is necessary to access its GitLab repository [11] and download a zip file with all the necessary files. After downloading the files, decompress the contents of the zip file in a directory. At this point, the configuration process can start. To do so, it will be helpful to follow these steps:

1. Open a command terminal.
2. Navigate to the directory where the solution files have been decompressed.
3. Build the Docker image from the Docker Compose file using the following command:

   `docker compose up --build`
4. Open a web browser and enter the URL http://localhost:3000/.

4.5 i4Q^DA – Services for Data Analytics

For installing the i4Q^DA Solution, access to the solution repository [12] and download a zip file with all the necessary artefacts. After the downloading process is complete, decompress the contents of the zip file in a directory. Then, the configuration process can begin. For this, the following steps must be followed:

1. Open a command terminal.
2. Navigate to the directory where the solution artefacts have been decompressed.
3. Execute the following command to add the environment variable `AIRFLOW_UID` along with its UID to the `.env` file, which will contain the environment variables needed to run the solution.

   `echo -e “AIRFLOW_UID=$(id -u)” > .env`
4. Build the Docker image from the Docker Compose file using the following command:

   `docker compose up --build`
5. Open a web browser and enter the URL http://localhost:3000/.

4.6 i4Q^DIT – Data Integration and Transformation Services

Note: As the i4Q^DIT Solution is highly dependent on the requirements of each i4Q Pilot, it is not possible to provide a general user manual. For this reason, the following installation and configuration guide refers to a single version of the i4Q^DIT Solution that offers some interactive user interface elements, allowing the user to experiment with a set of feature extraction techniques. This version of the i4Q^DIT Solution can be downloaded directly from its corresponding GitLab repository [13].

Note: To follow the steps indicated in this guide, it is recommended to have previously installed the Message Broker tool. For more information about the installation guide of this tool, please refer to its corresponding GitLab repository [14].

After accessing the i4Q^DIT Solution GitLab repository indicated above, the user must download a zip file with all the artefacts needed to install the solution. Once the download is complete, the user must follow these steps:

1. Decompress the contents of the zip file in a directory.
2. Include the necessary SSL certificates in the Certificates/client folder to establish a secure connection to the Message Broker (these certificates are generated during the installation of the Message Broker). The required certificates are:
   - Certificate Authority (PEM file).
   - Client SSL certificate (PEM file).
   - Private SSL key (PEM file).
3. Open a command terminal.
4. Navigate to the directory where the artefacts have been decompressed from the zip file and locate the solution's Docker compose file.
5. Execute the following command to build and run the containers defined in the file.

```
docker compose up -d
```

6. Open a web browser (preferably Google Chrome) and open the URL `http://localhost:8501/` to access the solution's web interface.

4.7 i4Q^DR – Data Repository

Note: The i4Q^DR Solution can be installed by downloading the zip file or by cloning the content from the GitLab repository [15].

The development of the i4Q^DR Solution is done by executing the bash script `run.sh`. This allows launching a set of scenarios from the different toolkits and a Trino instance configured to connect to these toolkits.

The deployment script accepts different parameters and options that allow to specify which data storage technologies and scenarios are to be deployed. For obtaining the command and options needed to run this script, two methods can be used:

- Solution deployment by using its graphical interface.
- Solution deployment by using the command terminal.

4.7.1 Deployment via graphical interface

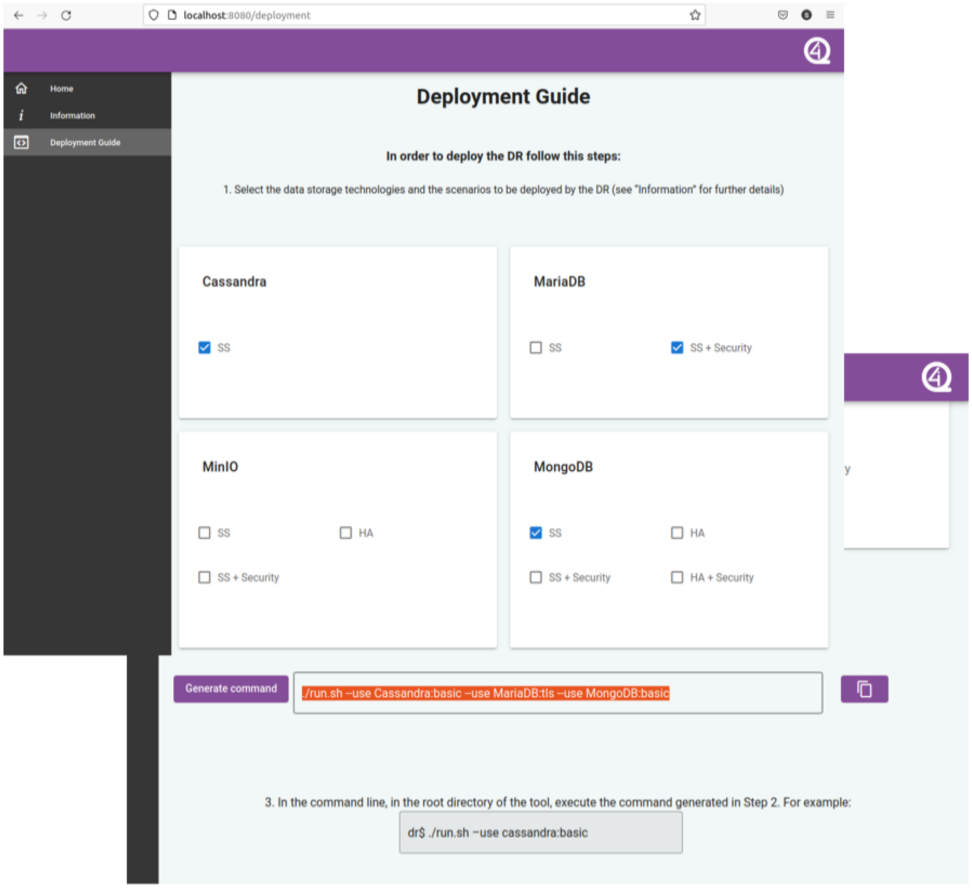


Figure 1.i4Q[DR] Solution – Deployment command generation via the GUI

For the solution to be deployed using its graphical interface, these steps must be followed:

1. Open a command terminal.
2. Navigate to the solution directory where the `runGUI.sh` script is located.
3. Run the script in charge of launching the solution graphical user interface using the following command:

   `./runGUI.sh`
4. Open a web browser and access the URL `http://localhost:8080`.
5. Click on the `Deployment Guide` option in the left-hand side menu.
6. Select the data storage technologies and scenarios to be deployed. This is done by selecting the checkboxes associated with the different technologies and scenarios.
7. Click on the `Generate command` button.
8. Click on the `Copy` button to copy the generated deployment command.
9. Open a new command terminal.
10. Navigate to the main directory of the solution.

11. Paste the command obtained from the browser into the terminal and execute it.

4.7.2 Deployment via command terminal

For the solution to be deployed using the command terminal, these steps must be followed:

1. Open a command terminal.
2. Navigate to the solution directory where the `run.sh` script is located.
3. To learn how to use the script, execute the following command:

   ```
   ./run.sh --help
   ```

4. To find out what scenarios are available, execute the following command:

   ```
   ./run.sh --list
   ```

5. To start a set of scenarios and a Trino instance, run a command similar to the one shown below. Specifically, this command allows to deploy: a Cassandra database using a "Single Server" scenario, a MongoDB database using the "High Availability" scenario, and a MySQL database with the "Single Server with Security" scenario.

> **Note**: For more detailed instructions on the deployment and configuration of the solution it is recommended to visit the official web documentation [16].

4.8 i4Q^DT – Digital Twin Simulation Services

> **Note**: The i4Q^DT Solution can be installed by downloading the zip file or by cloning the GitLab repository [17].

For the configuration of the Docker environment, these steps must be followed:

1. Open the docker-compose.yml file.
2. Indicate in `services/backend/volumes` in which folder the physics-based models, co-simulation files, STL models, data-driven data and data-driven models are going to be stored (the default folder or another).
3. Configure the ports for frontend and backend in `services/frontend/ports` and `services/backend/ports`, respectively. Then, save and close the file.
4. Open the `config.yml` file.
5. Define the backend IP address and port. Then, save and close the file.
6. Open the `config.js` file.
7. Define the backend IP address and port. Then, save and close the file.
8. Launch the application.
9. Open a new terminal in the folder where the `docker-compose.yml` file is located and run the following command for building the corresponding Docker images.

   ```
   docker-compose up –build
   ```

10. The application will be running as soon as the process finishes. The URL where the user interface is running will be shown in the terminal, which by default is `http://localhost:8082/`.

The application of this solution is general, as its main functionality is that of creating and simulating both data-driven models and physics-based models. The only consideration that must be taken into account is regarding the latter ones: uploaded FMU files must either contain a valid Linux binary dynamic library file or the source code so they can be automatically compiled by the library inside Docker.

4.9 i4Q^{EW} – Edge Workloads Placement and Deployment

Note: The step-by-step user manual can be found in the Gitlab repository [4]. For a combined demo with the i4Q^{LRT} Solution, please refer to the i4Q^{LRT} Solution GitLab repository [5]. Requirements include creation, import, attachment and controlling deployment locations via label-based policies.

Note: Main underlying technology used by this solution is ACM (Advanced Cluster Management), which is in charge of the heavy lifting of artefacts. Artefacts accepted includes Kubernetes-based structures.

4.10 i4Q^{IM} – Infrastructure Monitoring

Note: The i4Q^{IM} Solution can be downloaded directly from the associated GitLab repository [18].

Note: To follow the steps indicated in this guide, it is recommended to have previously installed the Message Broker tool. For more information about the installation guide of this tool, please refer to its corresponding GitLab repository [14].

After accessing the solution's GitLab repository, the user must download a zip file with all the artefacts needed to install the solution. Once the download is complete, the user must follow these steps:

1. Decompress the contents of the zip file in a directory.
2. Include the necessary SSL certificates in the Certificates/client folder to establish a secure connection to the Message Broker (these certificates are generated during the installation of the Message Broker). The required certificates are:
    - Certificate Authority (PEM file).
    - Client SSL certificate (PEM file).
    - Private SSL key (PEM file).

3. Open a command terminal.
4. Navigate to the directory where the artefacts have been decompressed from the zip file and locate the solution's Docker compose file.
5. Execute the following command to build and run the containers defined in the file.

   docker compose up -d

6. Open a web browser (preferably Google Chrome) and open the URL `http://localhost:8502/` to access the solution's web interface.

4.11 i4Q^LRT – Manufacturing Line Reconfiguration Toolkit

Note: This component has been designed to be used both with Docker and Kubernetes. For each of these technologies a different procedure to follow is detailed below.

4.11.1 Docker environment configuration

Docker-based installation can be achieved by running a `docker-compose` command. For each data processing algorithm, a separate Docker compose file is provided. For example, the following command starts an instance of the component.

```
docker-compose -f docker-compose.[Algorithm Name].yaml up
```

4.11.2 Kubernetes environment configuration

To deploy this solution in a Kubernetes cluster, helm files are prepared, which can be launched from commands. For example, the following command starts an instance of the Machine-Job Assignment Naive Algorithm.

```
helm install [Algorithm Name] ./ --namespace [Algorithm Name] --set defaultSettings.registrySecret=[registrySecret]
```

4.12 i4Q^PA – Prescriptive Analysis Tools

Note: The i4Q^PA Solution can be install by downloading the zip file or by cloning the GitLab repository [19].

For the configuration of the Docker environment, these steps must be followed:

1. If the files available for testing are to be used, move the folder `PA_files` to a working directory.
2. Open the docker-compose.yml file.
3. Indicate in `services/backend/volumes` in which folder the models, and custom evaluation functions, metrics and results are going to be stored (the `PA_files` folder's

working subdirectories or others).

4. FMU files must either contain a valid Linux binary dynamic library file or the source code so they can be automatically compiled by the library inside Docker. Then, save and clove the file.
5. Open the `config.yml` file.
6. Define the users custom predefined configuration (all users will start with the following configuration but will be able to change it in the app): `data_export`, `msg_broker`, `remote_storage`, and `configuration`.
7. Define simulations pre-defined configuration. Next, define the backend IP address and port. Then, save and close the file.
8. Launch the application.
9. Open a new terminal in the folder where the `docker-compose.yml` file is located and run the following command for building the corresponding Docker images.

```
docker-compose up --build
```

10. The application will be running as soon as the process finishes. The user interface will be accessible in the URL `http://localhost:8000` and, to access this service, the use of the Google Chrome browser is recommended.

4.13 i4Q^PQ – Data-Driven Continuous Process Qualification

Note: The i4Q^PQ Solution can be downloaded directly from its GitLab repository [20]. The solution can be deployed in two ways: via terminal and via Docker.

4.13.1 Deployment via the terminal

In order to deploy the solution via the terminal, these steps must be followed:

1. Open a command terminal.
2. Clone the solution's GitLab repository on the machine where it will be deployed by using the following command.

```
git clone https://gitlab.com/i4q/PQ.git
```

3. Navigate to the directory where the libraries and executable files are located using the following command.

```
cd PQ/subsystems/cont_proc
```

4. Install the dependencies needed to run the application using the following command.

```
pip install -r requirements.txt
```

5. Start the application by running this command.

```
streamlit run app.py
```

4.13.2 Deployment via Docker

For those who prefer containerisation or require a Docker-based environment, these steps must be followed for deploying the solution:

1. Open a command terminal.
2. Clone the solution's GitLab repository by using the following command.

   ```
   git clone https://gitlab.com/i4q/PQ.git
   ```

3. Navigate to the directory where the Docker compose is located with the following command.

   ```
   cd PQ/subsystems/cont_proc
   ```

4. Run the following command to build and run the containers defined in the Docker compose file, so that the dockerised version of the solution can be deployed.

   ```
   docker-compose up
   ```

4.14 i4Q^QD – Rapid Quality Diagnosis

Note: The i4Q^QD Solution can be downloaded directly from the associated GitLab repository [21].

Note: To follow the steps indicated in this guide, it is recommended to have previously installed the Message Broker tool. For more information about the installation of this tool, please refer to its corresponding GitLab repository [14].

After accessing the solution's GitLab repository, the user must download a zip file with all the artefacts needed to install the solution. Once the download is complete, the user must follow these steps:

1. Decompress the contents of the zip file in a directory.
2. Include the necessary SSL certificates in the `Certificates/client` folder, to establish a secure connection to the Message Broker (these certificates are generated during the installation of the Message Broker). The required certificates are:
   - Certificate Authority (PEM file).
   - Client SSL certificate (PEM file).
   - Private SSL key (PEM file).
3. Open a command terminal.
4. Navigate to the directory where the artefacts have been decompressed from the zip file and locate the solution's Docker compose file.
5. Execute the following command to build and run the containers defined in the file.

   ```
   docker compose up -d
   ```

6. Open a web browser (preferably Google Chrome) and open the URL `https://localhost:8503` to access the solution's web interface.

4.15 i4Q^{QE} – QualiExplore for Data Quality Factor Knowledge

Note: The i4Q^{QE} Solution consists of two main components: the Angular application and the database. For detailed instructions on how to deploy the application, it is recommended to consult the `qualiexplore` GitHub repository [22]. On the other hand, to consult the instructions for deploying the application and the database together in a dockerised way, it is recommended to visit the `qualiexplore-stack` GitHub repository [23].

In general terms, the steps to be followed for deploying the i4Q^{QE} Solution are as follows:

1. Build the i4Q^{QE} image from the `qualiexplore` repository.
2. Deploy the services using Docker Compose from the `qualiexplore-stack` repository.

Note: For modifying the default generated users, it is recommended to edit the content of the `users.json` file included in the `qualiexplore-stack` repository [24].

4.16 i4Q^{SH} – IIoT Security Handler

The main purpose of this solution is the generation of the Certification Authority (CA) so, in this context, once the data that the CA must have is specified and the CA is generated, the solution as such is not directly used. Instead, each solution that needs certificates generated by the i4Q^{SH} Solution will generate them using that CA.

4.17 i4Q^{TN} – Trusted Networks with Wireless and Wired Industrial Interfaces

4.17.1 Installation of the IWSN system

The complete installation of the IWSN subsystem, included in the i4Q^{TN} Solution, requires the compilation and installation of different software components in various hardware platforms. First, the firmware compilation and upload for the Wireless Sensor nodes, based on Contiki-NG OS, SDN-Wise and the needed developments to combine the use of Time-Slotted Channel Hopping (TSH) medium access method with the proper functions to support SDN scheduling rules and quality reports to the SDN controller. All the components require specific hardware platforms, and the subsystem cannot be virtualised or containerised, since the different platforms requires a direct connection with different physical interfaces, such as 6LoWPAN network interface or the local system configuration to set up the Wi-Fi AP or the Avahi mDNS services.

1. Uploading the firmware to the nodes

   To upload the firmware to the nodes it is necessary to set up the Contiki-NG Toolchain, either through a native installation (Linux) or using the official Docker images. The easiest and fastest way is to use the official Contiki-NG images. For further information regarding the native or Docker installation, please follow the steps of the official wiki [25].

```
docker pull contiker/contiki-ng
```

Instead of cloning the official repository [26], download the compressed zip file with the i4Q Contiki-NG and start Docker with the following options:

```
docker run --privileged --sysctl net.ipv6.conf.all.disable_ipv6=0 --
mount type=bind, source=<absolute-path-to-the-downloaded-i4q-contiki-
ng>,destination=/home/user/contiki-ng -e DISPLAY=$DISPLAY -v
/tmp/.X11-unix:/tmp/.X11-unix -v /dev/bus/usb:/dev/bus/usb -ti
contiker/contiki-ng
```

Finally, to build and upload the firmware, the IWSN node must be attached to the `contiki-ng` Docker container, and the serial device must be specified to properly upload the firmware to the IWSN node (the following example shows the /dev/ttyUSB0 serial device). The `TARGET=zoul` variable must be included, since the hardware platform developed is based on CC2538 TI transceiver and some other common components based on zoul platform pin assignment. To configure different node ids for each IWSN node, different hexadecimal numbers must be defined using the `NODEID` inline variable.

```
cd examples/i4qtn-iwsn/
sudo make TARGET=zoul MOTES=/dev/ttyUSB NODEID=x1 i4q-node.upload
```

In case any problem occurs, the firmware can also be uploaded using the SmartRF Flash Programmer 2 from Texas Instruments and the .hex firmware image of the i4q-node, using the TI XDS Debug Probe.

2. Start the SDN controller and the IWSN Gateway

The IWSN Gateway is based on a Single Board Computer (Raspberry Pi 4 or Tinker Board) that requires additional components to be able to interconnect with the 6LoWPAN IWSN nodes. Any mote with the CC2538 transceiver (Zolertia RE-Mote, Zolertia Firefly, Openmote or even the i4Q IWSN hardware developed) must be attached to the IWSN Gateway to define a new network interface to communicate with the IWSN nodes.

Since the IWSN Gateway requires different subcomponents to properly deploy the IWSN network, a bash script has been defined to schematically supervise the deployment process of each component of the IWSN Gateway. The official i4Q^{TN} repository only hosts the components of the SDN controller and the backend subcomponent of the complete platform.

```
sudo ./start.sh
```

This script supervises the deployment of each of the following subcomponents: mDNS Avahi service, 6LoWPAN network interface, local Wi-Fi Access Point, local MongoDB database, Java SDN Controller, the backend service and the frontend service.

4.17.2 Installation of the TSN system

The complete installation of a TSN system is not a straightforward task. It is more than just setting up the hardware with TSN IP, which is a first requirements for supporting TSN in the system. The setting up of such a system is application dependent, as it depends on the currently running application what kind of TSN schedule is required for sending messages. As this cannot be described in detail (as it is completely application specific), it is advisable for the end user to get into contact with the technology provider (i.e., TTTEch Industrial) for support on setting up the overall system.

4.18 Message Broker

Note: The Message Broker tool can be downloaded directly from the associated GitLab repository [14].

After accessing the tool's GitLab repository, the user must download a zip file with all the artefacts needed to install the tool. Once the download is complete, the user must follow these steps:

1. Decompress the contents of the zip file in a directory.
2. Open a command terminal.
3. Navigate to the directory where the `config.yml` file is located and define the following properties:
   - Host machine IP address (e.g., `160.40.56.80`).
   - Names of i4Q Solutions using SSL certificates (e.g., "`IM QD DIT`").
   - Name of the CA certificate and key (e.g., `i4q_ca-cert` and `i4q_ca-key`).
4. Run the following command to install the tool.

   ```
   bash start_service.sh
   ```
5. Once the installation is complete, the Kafka endpoints of the Message Broker will be accessible through the following URLs, corresponding to the 3 deployed Kafka nodes:
   - `http://<HOST_MACHINE_IP>:9092`
   - `http://<HOST_MACHINE_IP>:9093`
   - `http://<HOST_MACHINE_IP>:9094`

Note: For a more comprehensive and detailed guide for the configuration, installation, and usage of the Message Broker, it is recommended to consult the manual with usage examples [27] or the manual with detailed instructions on the configuration and deployment of the tool [28], both of which can be found in the tool's GitLab repository.

5. Solutions user manual

Once the installation and configuration of the i4Q Solutions is complete, the user needs to have an up-to-date guide or user manual that explains step-by-step how each i4Q Solution works and what each of the configuration parameters means. To this end, this section provides an updated user manual, or indicates where a detail manual on the solution's functioning can be found. New functionalities that have not been explain so far can also be described.

5.1 i4Q^{AD} – Analytics Dashboard

This solution is built based on Apache Superset, which is an open-source software cloud-native application for data exploration and data visualisation. Guidelines and the user manual for Apache Superset can be found on the Apache Superset official website [29].

For creating a data streaming pipeline for visualisation, Apache Druid is used as the middleware. The latest guidelines for the pipeline configuration can be found on the Apache Druid official website [30].

No significant modifications have been made to the user manual of this solution. For more information, the official version of this manual can be found in the web documentation [31].

5.2 i4Q^{AI} – AI Models Distribution to the Edge

This solution has been developed using ACM (Advanced Cluster Management) technology as a source of inspiration. This technology is in charge of doing the heavy lifting for the artefacts, which can include Kubernetes-based structures. For more information on this technology, please visit the official Red Hat documentation [32].

No significant modifications have been made to the user manual of this solution. For more information, the official version of this manual can be found in the web documentation [33].

5.3 i4Q^{BC} – Blockchain Traceability of Data

With respect to the i4Q^{BC} Solution user manual, there have been no significant modifications that imply adding or deleting content from the previous version. The official version of this manual can be found in the web documentation [34]. Other guides on the solution deployment or configuration can also be found in the GitLab repository [35].

5.4 i4Q^{BDA} – Big Data Analytics Suite

Concerning the user manual of the i4Q^{BDA} Solution, no relevant modifications have been made that imply changes to the content of the previous version. The official version of this manual can be found in the web documentation [36]. Other guidelines on the deployment or configuration of the solution can also be found in the GitLab repository [37].

5.5 i4Q^{DA} – Services for Data Analytics

As for the i4Q^{DA} Solution user's manual, modifications have been made to the user interface and the available operators. These changes can be consulted in the solution's GitLab repository [12].

A more complete version of the solution's operation and other less technical aspects can also be found in the solution's web documentation [38].

5.6 i4Q^DIT – Data Integration and Transformation Services

The implementation and deployment of the i4Q^DIT Solution varies significantly between the different i4Q Pilots. It is therefore complex to provide general instructions on its functionalities, as these will depend on the pilot in which the solution is intended to be deployed.

Most of the operations of the i4Q^DIT Solution are automated, as its purpose is to prepare in real-time the manufacturing data that is used in the subsequent analysis performed by the other i4Q Solutions for each pilot.

In order to illustrate the new interactive functionalities implemented in the solution, a series of images of the graphical interface designed for this solution in the FIDIA pilot are shown below.

5.6.1 Data source selection

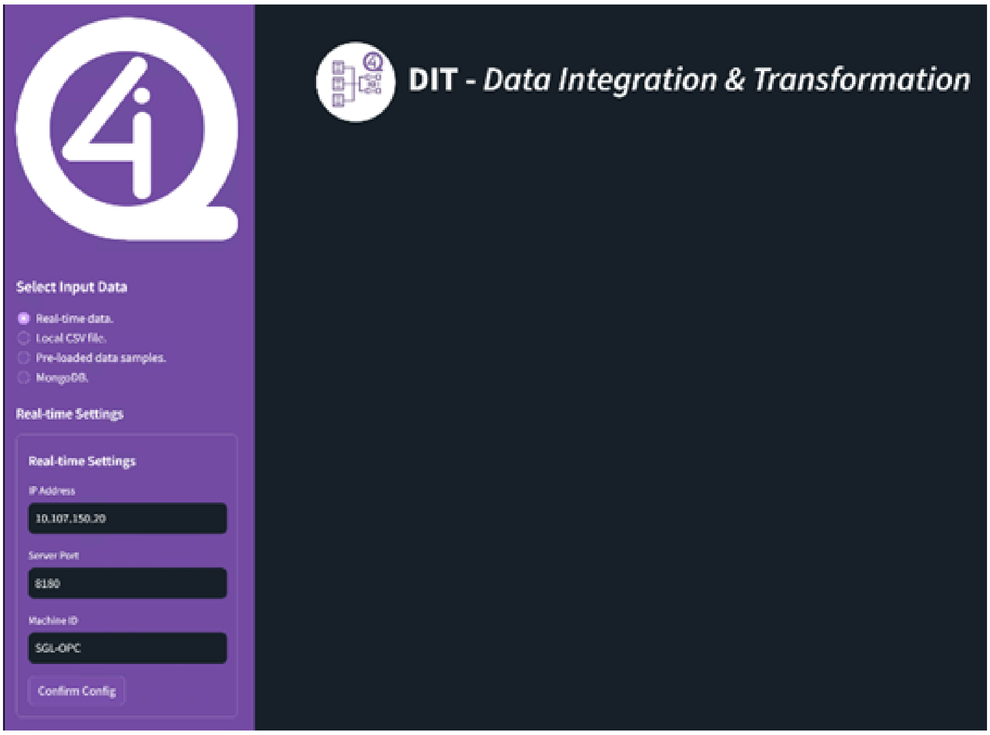


Figure 2. i4Q^DIT Solution – Data source selection functionality

The user can load data from different data sources:

- **Real-time data**. The user can enter real manufacturing data into the platform using the API provided by FIDIA, specifying the IP address, port, and machine id. The data is then processed in the background and sent through the Message Broker using the *DTI_topic_1* topic.
- **Local CSV files**. The user can upload a CSV file with the data to be processed.

- **Preloaded data samples**. Allows the user to test the solution with a set of data already uploaded by the user.
- **MongoDB**. Allows the user to import data from a MongoDB database and data collection managed by the i4Q^{DR} Solution.

5.6.2 Data preprocessing

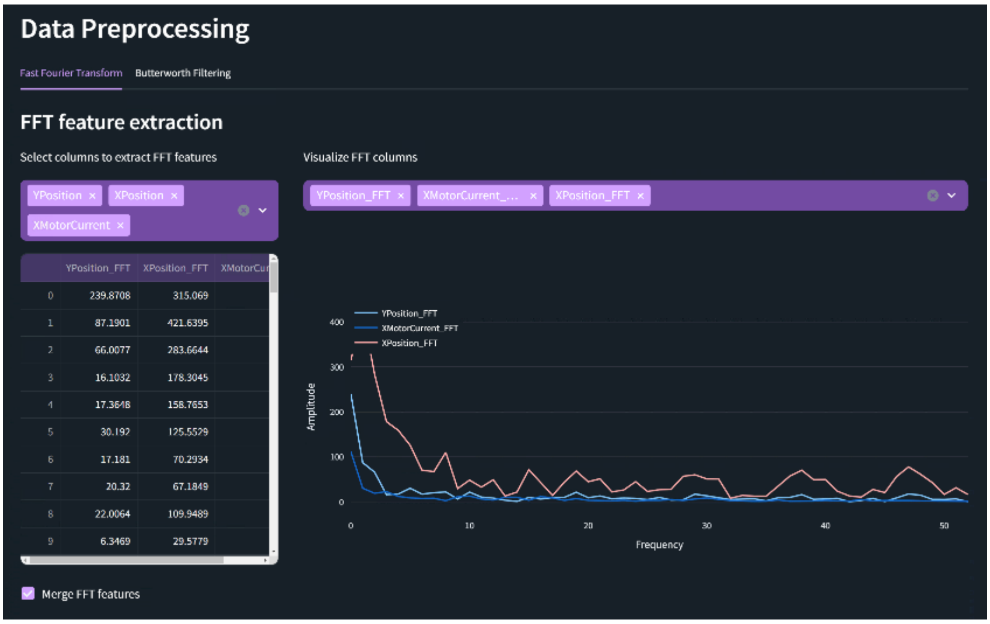


Figure 3. i4Q^{DIT} Solution – Data preprocessing functionality

Selecting a different data source, the user can load static data and apply a set of preprocessing techniques to enrich them. By choosing some of the sensors, it is possible to extract new information through FFT and Butterworth filtering and finally merge the new features into the original dataset. The newly extracted sensor features are also being presented via line-charts.

5.6.3 Data provision and storage

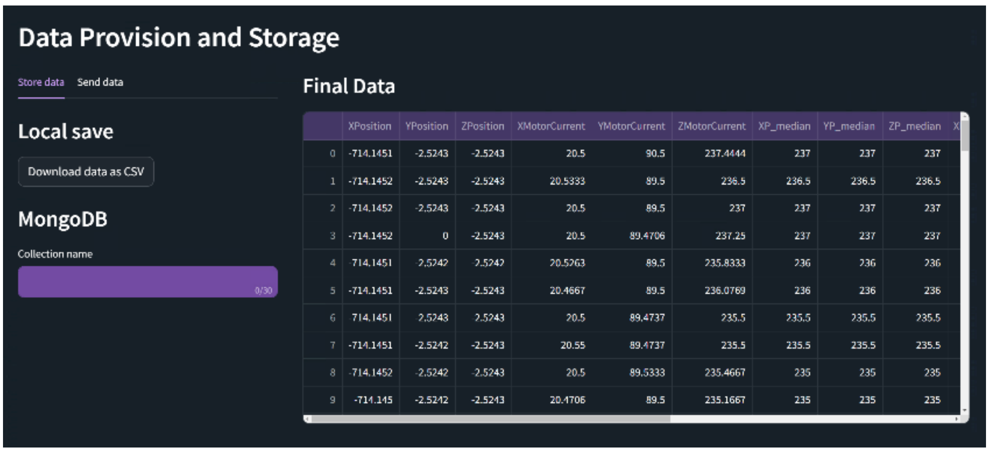


Figure 4. i4Q^{DIT} Solution – Data provision and storage functionality

Finally, the user has the options of downloading the final prepared dataset in a CSV format, save it in a MongoDB data collection provided by the i4Q^{DR} Solution or send it via the Message Broker.

Once the installation and deployment of the solution is completed, the web service and the aforementioned functionalities will be available at the URL `http://localhost:8501/`

For other less technical aspects of the i4Q^{DIT} Solution, it is recommended to visit the solution's web documentation [39].

5.7 i4Q^{DR} – Data Repository

With regard to the i4Q^{DR} Solution user manual, it has not suffered any modification since the last published version. This last latest official version of the manual can be consulted through the web documentation [16].

5.8 i4Q^{DT} – Digital Twin Simulation Services

With respect to the i4Q^{DT} Solution user manual, there have been no significant modifications that entail the addition or deletion of content from the previous version. The official version of this manual can be found in the web documentation [40].

5.9 i4Q^{EW} – Edge Workloads Placement and Deployment

As for the i4Q^{EW} Solution user's manual, there have been no major changes that imply the addition or deletion of content from the previous version. The latest version of this manual can be found in the web documentation [41]. Other guidelines on the solution deployment or configuration can also be found in the GitLab repository [4].

5.10 i4Q^{IM} – Infrastructure Monitoring

With respect to the latest version of the i4Q^{IM} Solution user manual, several changes have been registered, which are given by a series of new interactive functionalities. To present these new functions implemented in the solution, a series of images of the user interface are shown below, together with a brief explanation of each one of them.

5.10.1 User role selection

Based on the expertise of the user, the tool adjusts its functionalities. The **data scientist role** allows for historical data analysis and the training of new Machine Learning algorithms. On the other hand, the **operator role** is oriented towards real-time analytics.

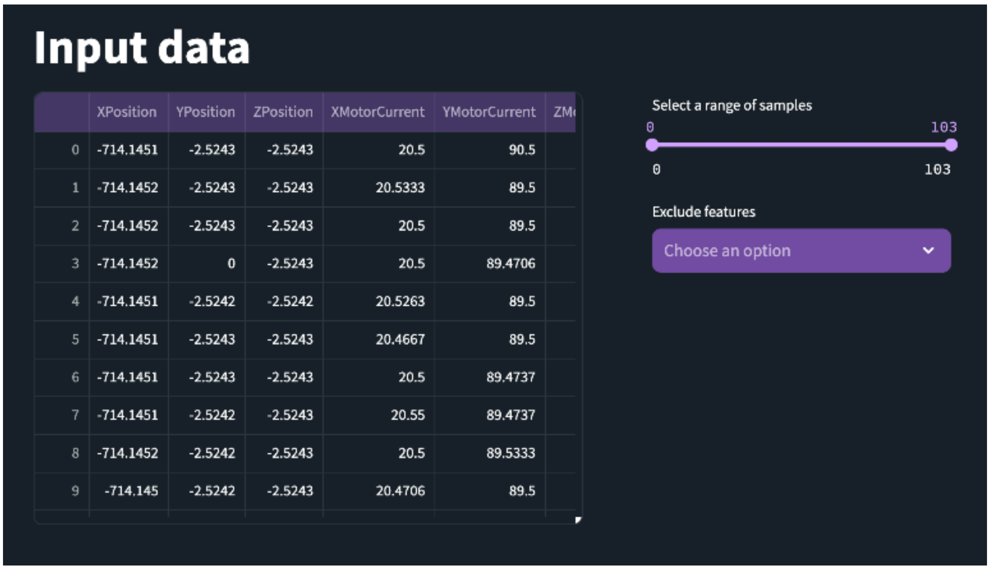
5.10.2 Data source selection

The user can upload data by selecting from a set of different data input options. These options include:

- **Pre-loaded data samples**. Allows the user to test the solution with a set of data already uploaded by the user.
- **Local CSV files**. The user can upload a CSV file with the data to be processed.
- **MongoDB**. Allows the user to import data from a MongoDB database and data collection managed by the i4Q^{DR} Solution.
- Message Broker.

Figure 5. i4Q[IM] Solution – Data source selection

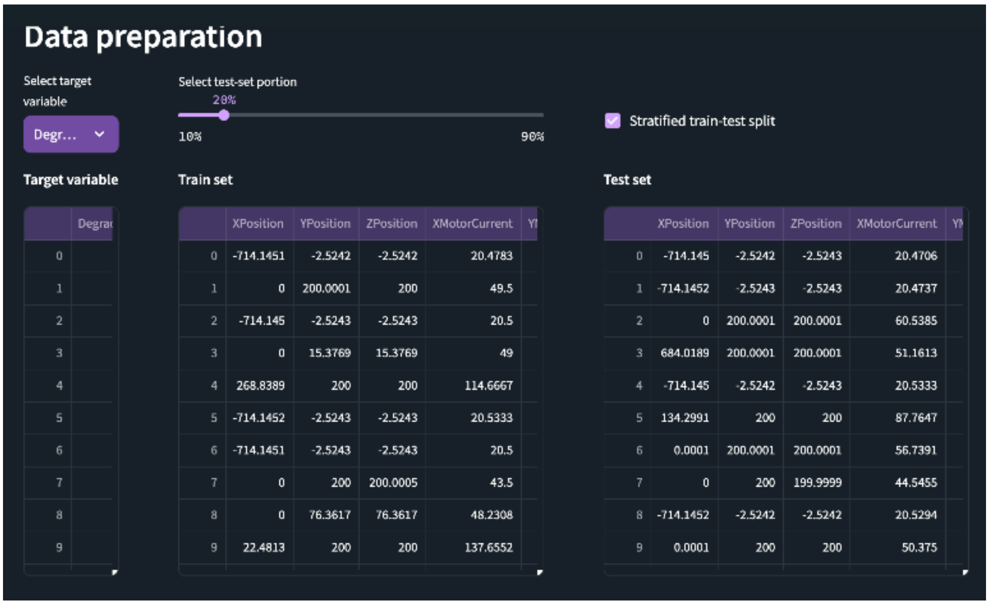
In the **Data Scientist** tab, once the data has been loaded, a table is presented containing the entirety of the dataset, along with a slider to adjust the number of data samples.



	XPosition	YPosition	ZPosition	XMotorCurrent	YMotorCurrent	ZM
0	-714.1451	-2.5243	-2.5243	20.5	90.5	
1	-714.1452	-2.5243	-2.5243	20.5333	89.5	
2	-714.1452	-2.5243	-2.5243	20.5	89.5	
3	-714.1452	0	-2.5243	20.5	89.4706	
4	-714.1451	-2.5242	-2.5242	20.5263	89.5	
5	-714.1451	-2.5243	-2.5243	20.4667	89.5	
6	-714.1451	-2.5243	-2.5243	20.5	89.4737	
7	-714.1451	-2.5242	-2.5243	20.55	89.4737	
8	-714.1452	-2.5242	-2.5243	20.5	89.5333	
9	-714.145	-2.5242	-2.5243	20.4706	89.5	

Figure 6. i4Q[IM] Solution – Input data and data samples selection

The user can then select the target variable that will be used in the training process of the detection algorithm. Also, an option to adjust the ratio of training and testing data samples is provided with or without the user of a stratified splitting strategy.



	XPosition	YPosition	ZPosition	XMotorCurrent
0	-714.1451	-2.5242	-2.5242	20.4783
1	0	200.0001	200	49.5
2	-714.145	-2.5243	-2.5243	20.5
3	0	15.3769	15.3769	49
4	268.8389	200	200	114.6667
5	-714.1452	-2.5243	-2.5243	20.5333
6	-714.1451	-2.5243	-2.5243	20.5
7	0	200	200.0005	43.5
8	0	76.3617	76.3617	48.2308
9	22.4813	200	200	137.6552

	XPosition	YPosition	ZPosition	XMotorCurrent
0	-714.145	-2.5242	-2.5243	20.4706
1	-714.1452	-2.5243	-2.5243	20.4737
2	0	200.0001	200.0001	60.5385
3	684.0189	200.0001	200.0001	51.1613
4	-714.145	-2.5242	-2.5243	20.5333
5	134.2991	200	200	87.7647
6	0.0001	200.0001	200.0001	56.7391
7	0	200	199.9999	44.5455
8	-714.1452	-2.5242	-2.5242	20.5294
9	0.0001	200	200	50.375

Figure 7. i4Q^{IM} Solution – Data preparation

Finally, the user can either choose to train a new AI detection model or select one from the available pre-trained models. Then, the user can observe the samples in which a machine malfunction/degradation has been detected along with the magnitude of contribution that each sensor has towards the predictions of the model.

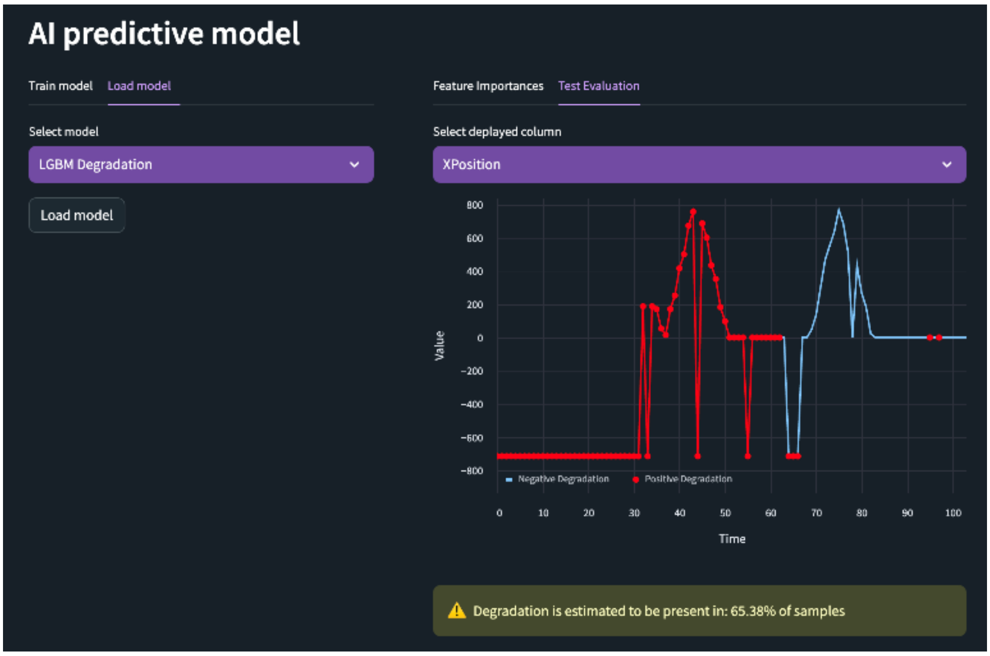


Figure 8. i4Q^{IM} Solution – Model training and results evaluation

In the **Operator's** tab, the use can select one of the pre-trained AI models and specify the Kafka server and topic to ingest real-time machine sensor data into the platform.

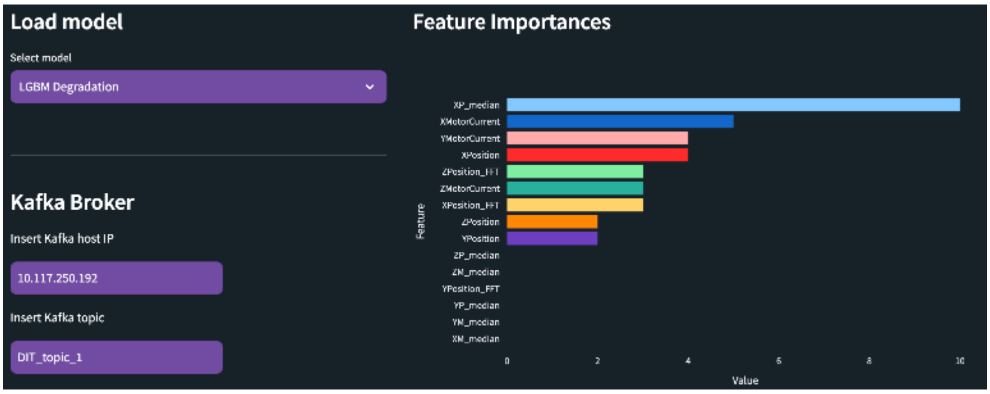


Figure 9. i4Q^{IM} Solution – Real-time data production using Apache Kafka

By enabling the stream-analytics, the selected predictive model is evaluating the presence of general machine malfunctions or specific component degradation in real-time and presents the results through live charts.

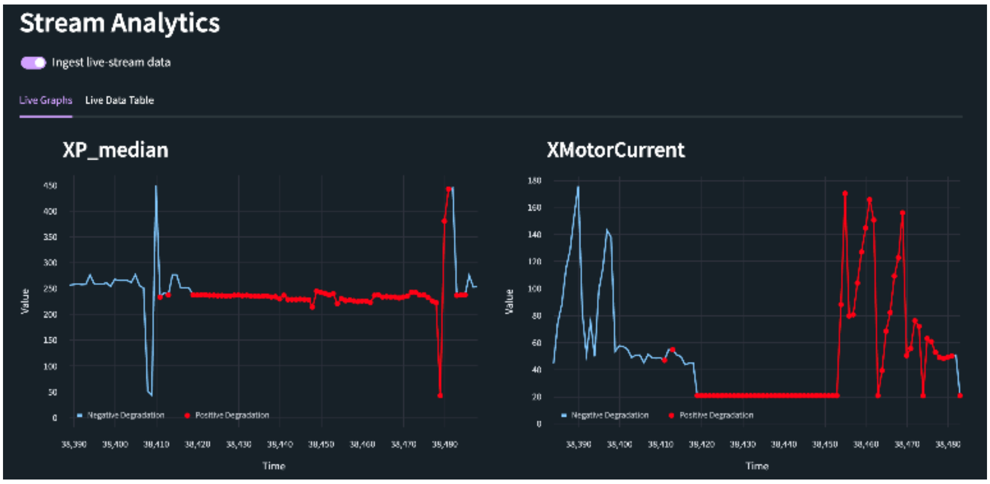


Figure 10. i4Q^{IM} Solution – Stream analytics in real-time

5.11 i4Q^{LRT} – Manufacturing Line Reconfiguration Toolkit

Regarding the user manual of the i4Q^{LRT} Solution, no relevant modifications have been made that imply changes to the content of the previous version. The official version of this manual can be found in the web documentation [42].

5.12 i4Q^{PA} – Prescriptive Analysis Tools

With respect to the i4Q^{PA} Solution user manual, there have been no significant modifications that entail the addition or deletion of content from the previous version. The official version of this manual can be found in the web documentation [43].

5.13 i4Q^{PQ} – Data-Driven Continuous Process Qualification

As for the user manual of the i4Q^{PQ} Solution, no major modifications have been made that imply changing the content of the previous document. The official version of this solution manual can be found in the web documentation [44].

5.14 i4Q^{QD} – Rapid Quality Diagnosis

With respect to the latest version of the i4Q^{QD} Solution user manual, several modifications have been registered, which are given by a series of new interactive functionalities. To present these new functions implemented in the solution, a series of images of the user interface are shown below, together with a brief explanation of each one of them.

5.14.1 User role selection

Based on the expertise of the user, the tool adjusts its functionalities. The **data scientist** role allows for historical data analysis and the training of new Machine Learning algorithms. On the other hand, the **operator data** is oriented towards real-time analytics.

5.14.2 Data source selection

The user can upload data by selecting from a set of different data input options. These options include:

- **Pre-loaded data samples**. Allows the user to test the solution with a set of data already uploaded by the user.
- **Local CSV files**. The user can upload a CSV file with the data to be processed.
- **MongoDB**. Allows the user to import data from a MongoDB database and data collection managed by the i4Q^{DR} Solution.
- **Message Broker**.

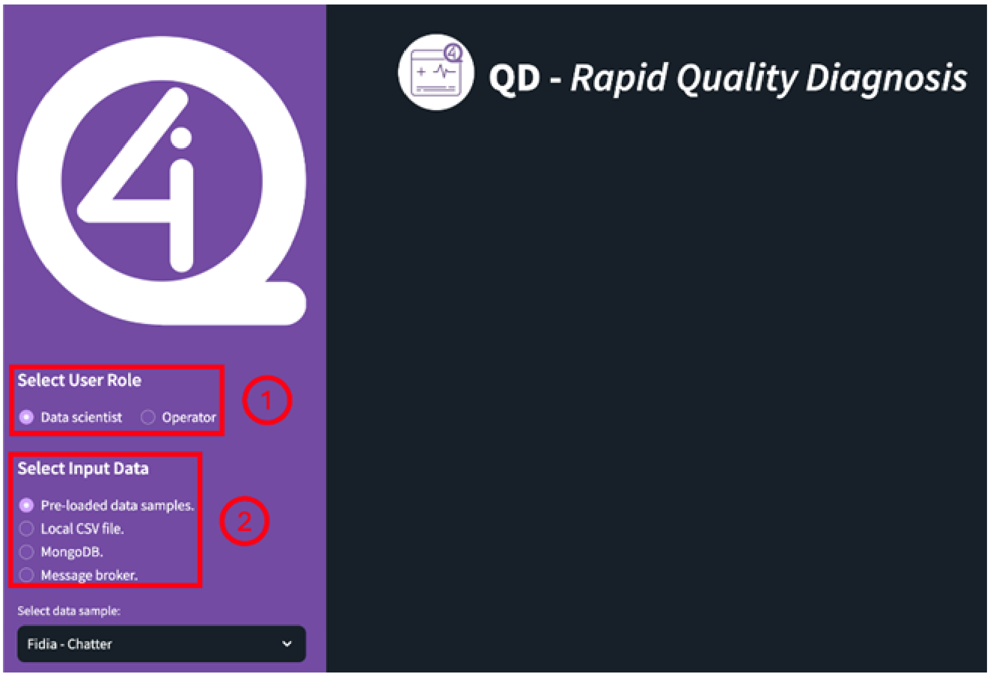


Figure 11. i4Q^{QD} Solution – Data source selection

In the **Data Scientist** tab, once the data has been loaded, a table is presented containing the entirety of the dataset, along with a slider to adjust the number of data samples.

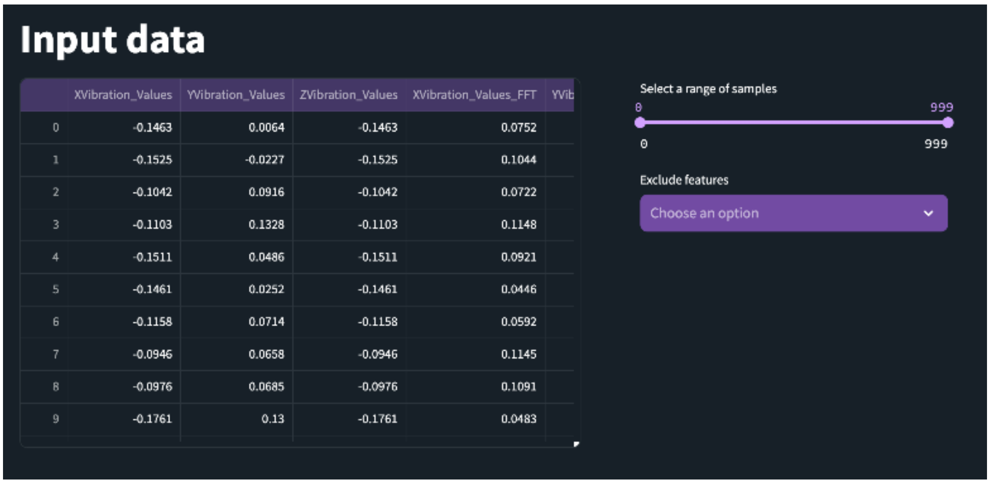


Figure 12. i4Q[QD] Solution – Input data and data samples selection

The user can then select the target variable that will be used in the training process of the detection algorithm. Also, an option to adjust the ratio of training and testing data samples is provided with or without the user of a stratified splitting strategy.

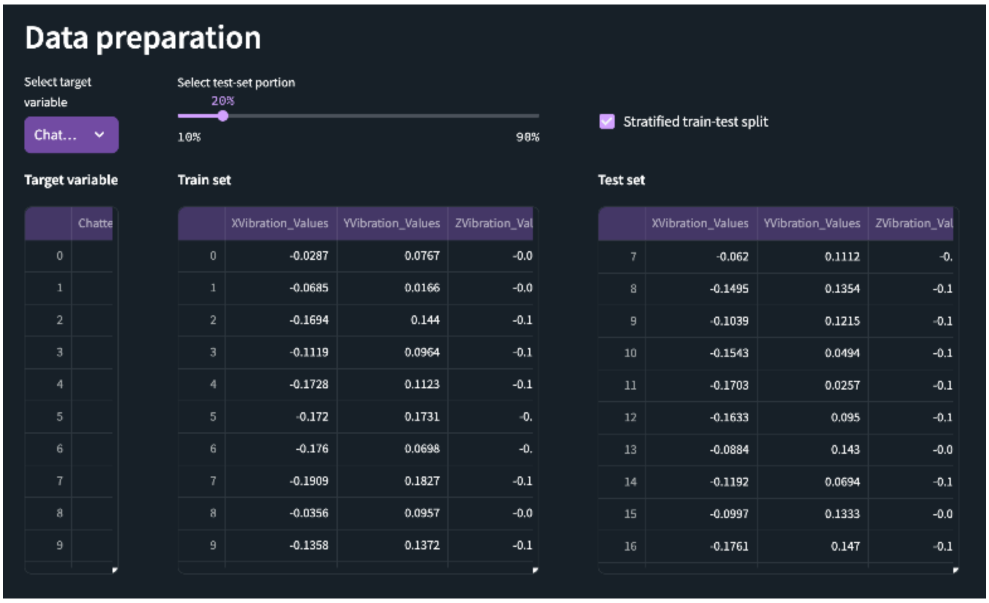


Figure 13. i4Q[QD] Solution – Data preparation

Finally, the user can either choose to train a new AI detection model or select one from the available pre-trained models. Then, the user can observe the samples in which a product defect or a problem during the manufacturing process has been detected, along with the magnitude of contribution that each sensor has towards the predictions of the model.

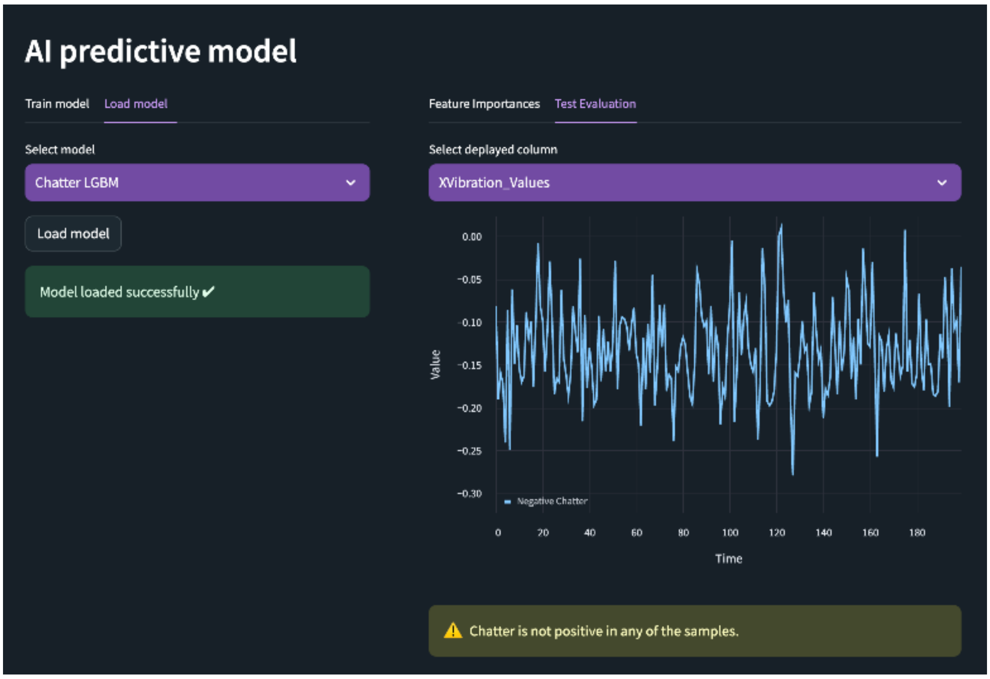


Figure 14. i4Q[QD] Solution – Model training and results evaluation

In the **Operator's** tab, the use can select one of the pre-trained AI models and specify the Kafka server and topic to ingest real-time machine sensor data into the platform.

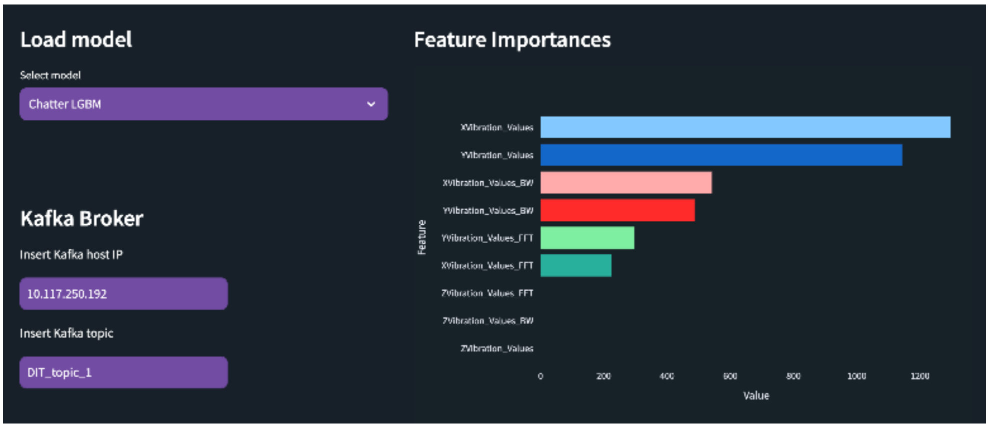


Figure 15. i4Q[QD] Solution – Real-time data production using Apache Kafka

By enabling the stream-analytics, the selected predictive model is evaluating the presence of a product defect or a problem during the manufacturing process in real-time and presents the results through live charts.

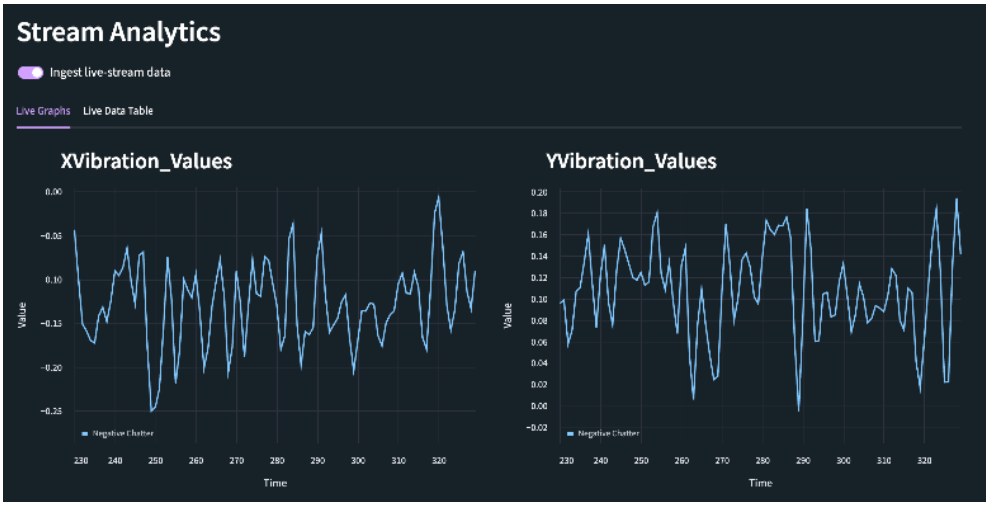


Figure 16. i4Q^{QD} Solution – Stream analytics in real-time

5.15 i4Q^{QE} – QualiExplore for Data Quality Factor Knowledge

The i4Q^{QE} Solution user manual has not experienced any changes since the last version. More details of the official version of this solution manual can be found in the web documentation [45].

5.16 i4Q^{SH} – IIoT Security Handler

Regarding the user manual of the i4Q^{SH} Solution, there have been no significant modifications that imply the addition or deletion of content from the last version. This last version of the official solution manual is available for consultation in the web documentation [46].

5.17 i4Q^{TN} – Trusted Networks with Wireless and Wired Industrial Interfaces

The i4Q^{TN} Solution user manual has not experienced any modifications since the last version. This last version of the official solution manual is available for review in the web documentation [47].

5.18 Message Broker

The Message Broker tool has two different versions of the user manual. One in PDF format more focused on usage examples, which can be accessed through this section of the GitLab repository [27]. Furthermore, a more extensive version with explanations of the steps to follow to deploy the solution can be found in this section of the same GitLab repository [28].

6. Conclusions

This document contains all the information regarding the i4Q Solutions user manual necessary for the users to download, deploy, install, and configure the solutions they are interested in using without the need for prior knowledge of the i4Q Project or the i4Q Solutions.

In the first place, a general description of each i4Q Solution has been presented, together with its main characteristics and the professional profile at which each of the solutions is aimed. In this way, it is intended that the user can acquire a general idea of each of the i4Q Solutions.

Afterwards, an analysis of the hardware and software requirements necessary to deploy and operate correctly each of the i4Q Solutions has been carried out. These requirements refer to each of the i4Q Solutions separately, so in case the user wants or needs to install more than one i4Q Solution, it is possible that the amount of some resources (e.g. available disk storage) may be higher. In addition, the calculation of the storage needed for each solution is estimated from the size of its associated resources, so extra free space should be available for those solutions that perform data storage using one or more databases or other data storage technologies (e.g. i4Q^{DR} Solution or Message Broker).

Next, a brief guide to the steps to follow for installing and configuring the different i4Q Solutions has been provided. This section is intended to be a summary of the content provided in the i4Q Project web documentation, as replicating all the content here would be useless. It is therefore recommended to visit the web documentation of the solution in question for more detailed information on how to configure certain aspects or scenarios.

At the end, the new functionalities implemented in some of the i4Q Solutions, that have not been included yet in the official web documentation of the i4Q Project, are explained. For those solutions that have not undergone modifications with respect to the content of the web documentation, it is recommended to consult the page corresponding to the i4Q Solution in question, as it provides a more detailed explanation of how to use each of the solutions, also including some screenshots of the interface that can be helpful to better understand the use of the solution.

References

[1] “i4Q Solutions”. i4Q Project web documentation. [Online]. Available: https://i4q.upv.es/

[2] S. Wellsandt. “D3.9 – Data Quality Guidelines v2,” BIBA – Bremer Institut für Produktion und Logistik GmbH, Bremen, Germany, Dec. 2022. [Online]. Available: https://knowledgebiz.sharepoint.com/:w:/r/sites/i4Q/_layouts/15/Doc.aspx?sourcedoc=%7BA0E3467C-C71C-4566-9D97-2A7690D1A0E2%7D&file=i4Q_Deliverable_D3.9_v0.4.docx

[3] “i4Q Analytics Dashboard”. i4Q Project GitLab repository. [Online]. Available: https://gitlab.com/i4q/AD

[4] “Importing a K8s cluster as a Managed-Cluster on a RHACM Instance”. i4Q Project GitLab repository. [Online]. Available: https://gitlab.com/i4q/LRT/-/tree/acm-dev/ACM/attach-cluster

[5] “ACM Demo Application – SKTime”. i4Q Project GitLab repository. [Online]. Available: https://gitlab.com/i4q/LRT/-/tree/acm-dev/ACM/demo-app

[6] “orionbcdb/orion-server – Docker Image”. Docker Hub. [Online]. Available: https://hub.docker.com/r/orionbcdb/orion-server

[7] “Container Registry”. i4Q Project GitLab repository. [Online]. Available: https://gitlab.com/groups/i4q/-/container_registries

[8] “i4Q^{BC} Solution”. i4Q Project GitLab repository. [Online]. Available: https://gitlab.com/i4q/bc

[9] “Launching a Orion Node > Orion Docker”. Hyperledger Orion. [Online]. Available: https://labs.hyperledger.org/orion-server/docs/getting-started/launching-one-node/docker/

[10] “Launching a Orion Node > Creating Crypto Material”. Hyperledger Orion. [Online]. Available: https://labs.hyperledger.org/orion-server/docs/getting-started/launching-one-node/crypto-materials/

[11] “i4Q^{BDA} Solution”. i4Q Project GitLab repository. [Online]. Available: https://gitlab.com/i4q/bda

[12] “i4Q Services for Data Analytics Documentation”. i4Q Project GitLab repository. [Online]. Available: https://gitlab.com/i4q/i4q-da

[13] “9-i4Q_DIT”. i4Q Project GitLab repository. [Online]. Available: https://gitlab.com/i4q/dit

[14] “i4Q – Message Broker”. i4Q Project GitLab repository. [Online]. Available: https://gitlab.com/i4q/message-broker

[15] ‘i4Q / DR · GitLab’. Accessed: Apr. 25, 2024. [Online]. Available: https://gitlab.com/i4q/dr

[16] “i4Q Data Repository (i4QDR)”. i4Q Project web documentation. [Online]. Available: https://i4q.upv.es/8_i4Q_DR/index.html

[17] “i4Q^{DT} Solution”. i4Q Project GitLab repository. [Online]. Available: https://gitlab.com/i4q/dt

[18] "i4Q Infrastructure Monitoring". i4Q Project GitLab repository. [Online]. Available: https://gitlab.com/i4q/15-i4q_im

[19] "i4Q Prescriptive Analysis Tools". i4Q Project GitLab repository. [Online]. Available: https://gitlab.com/i4q/pa

[20] "Continuous Process Qualification User Interface". i4Q Project GitLab repository. [Online]. Available: https://gitlab.com/i4q/PQ/-/tree/main/subsystems/cont_proc

[21] "i4Q Rapid Quality Diagnosis". i4Q Project GitLab repository. [Online]. Available: https://gitlab.com/i4q/18-i4q_qd

[22] S. Wellsandt. "QualiExplore Angular Module as a standalone app". GitHub repository s-wel/qualiexplore. [Online]. Available: https://github.com/s-wel/qualiexplore

[23] S. Wellsandt. "QualiExplore Stack with MongoDB integration (WIP)". GitHub repository s-wel/qualiexplore-stack. [Online]. Available: https://github.com/s-wel/qualiexplore-stack/tree/i4q

[24] S. Wellsandt. "File users.json from the qualiexplore-stack repository". GitHub repository s-wel/qualiexplore-stack. [Online]. Available: https://github.com/s-wel/qualiexplore-stack/blob/i4q/mongo_seed/users.json

[25] "Contiki-NG > Getting Started > Docker". Contiki-NG documentation. [Online]. Available: https://docs.contiki-ng.org/en/develop/doc/getting-started/Docker.html

[26] "Contiki-NG: The OS for Next Generation IoT Devices". GitHub repository contiki-ng/contiki-ng. [Online]. Available: https://github.com/contiki-ng/contiki-ng

[27] S. Paraschos. "i4Q – Message Broker user manual with usage examples". i4Q Project GitLab repository. [Online]. Available: https://gitlab.com/i4q/message-broker/-/blob/Generic-Pilot-Version/Message_Broker_-_User_manual.pdf

[28] "i4Q – Message Broker configuration and deployment". i4Q Project GitLab repository. [Online]. Available: https://gitlab.com/i4q/message-broker/-/blob/main/README.md

[29] "Introduction > What is Apache Superset?". Apache Superset documentation. [Online]. Available: https://superset.apache.org/docs/intro/

[30] "Getting started > Introduction to Apache Druid". Apache Druid documentation. [Online]. Available: https://druid.apache.org/docs/latest/design/index.html

[31] "i4Q – Analytics Dashboard". i4Q Project web documentation. [Online]. Available: https://i4q.upv.es/12_i4Q_AD/index.html

[32] "Red Hat Advanced Cluster Management for Kubernetes". RedHat.com. [Online]. Available: https://www.redhat.com/en/technologies/management/advanced-cluster-management

[33] "i4Q AI Models Distribution to the Edge". i4Q Project web documentation. [Online]. Available: https://i4q.upv.es/13_i4Q_AI/index.html

[34] "i4Q Blockchain Traceability of Data". i4Q Project web documentation. [Online]. Available: https://i4q.upv.es/3_i4Q_BC/index.html

[35] "i4Q Blockchain Traceability of Data". i4Q Project Gitlab repository. [Online]. Available: https://gitlab.com/i4q/Documentation/-/tree/master/documentation/source/3_i4Q_BC

[36] "i4Q Big Data Analytics Suite". i4Q Project web documentation. [Online]. Available: https://i4q.upv.es/11_i4Q_BDA/index.html

[37] "i4Q Big Data Analytics Suite". i4Q Project GitLab repository. [Online]. Available: https://gitlab.com/i4q/Documentation/-/tree/master/documentation/source/11_i4Q_BDA

[38] "i4Q Services for Data Analytics". i4Q Project web documentation. [Online]. Available: https://i4q.upv.es/10_i4Q_DA/index.html

[39] "i4Q Data Integration and Transformation Services". i4Q Project web documentation. [Online]. Available: https://i4q.upv.es/9_i4Q_DiT/index.html

[40] "i4Q Digital Twin". i4Q Project web documentation. [Online]. Available: https://i4q.upv.es/16_i4Q_DT/index.html

[41] "i4Q Edge Workloads Placement and Deployment". i4Q Project web documentation. [Online]. Available: https://i4q.upv.es/14_i4Q_EW/index.html

[42] "i4Q Manufacturing Line Reconfiguration Toolkit". I4Q Project web documentation. [Online]. Available: https://i4q.upv.es/21_i4Q_LRT/index.html

[43] "i4Q Prescriptive Analysis Tools". i4Q Project web documentation. [Online]. Available: https://i4q.upv.es/19_i4Q_PA/index.html

[44] "i4Q Data-Driven Continuous Process Qualification". i4Q Project web documentation. [Online]. Available: https://i4q.upv.es/17_i4Q_PQ/index.html

[45] "i4Q QualiExplore for Data Quality Factor Knowledge". i4Q Project web documentation. [Online]. Available: https://i4q.upv.es/2_i4Q_QE/index.html

[46] "i4Q Solution IIoT Security Handler (i4QSH)". i4Q Project web documentation. [Online]. Available: https://i4q.upv.es/6_i4Q_SH/index.html

[47] "i4Q Trusted Networks with Wireless and Wired Industrial Interfaces". i4Q Project web documentation. [Online]. Available: https://i4q.upv.es/4_i4Q_TN/index.html