# D4.15 – i4Q Infrastructure Monitoring v2

WP4 – BUILD: Manufacturing Data Analytics for Manufacturing Quality Assurance

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 | 36 months |
| PROJECT URL | https://www.i4q-project.eu/ | | |
| DELIVERABLE | D4.15 – i4Q Infrastructure Monitoring | | |
| WORK PACKAGE | WP4 – BUILD: Manufacturing Data Analytics for Manufacturing Quality Assurance | | |
| DATE OF DELIVERY | CONTRACTUAL 31-Dec-2022 | ACTUAL | 30-Dec-2022 |
| NATURE | Report | DISSEMINATION LEVEL | Public |
| LEAD BENEFICIARY | CERTH | | |
| RESPONSIBLE AUTHOR | Spyridon Paraschos, Myrsini Ntemi, Georgia Apostolou, Ilias Gialampoukidis (CERTH) | | |
| CONTRIBUTIONS FROM | - | | |
| TARGET AUDIENCE | 1) i4Q Project partners; 2) industrial community; 3) other H2020 funded projects; 4) scientific community | | |
| DELIVERABLE CONTEXT/ DEPENDENCIES | This document extends the implementation phase of the Infrastructure Monitoring solution (i4Q^IM) as described in *D4.7 i4Q Infrastructure Monitoring*. | | |
| EXTERNAL ANNEXES/ SUPPORTING DOCUMENTS | None | | |
| READING NOTES | None | | |
| ABSTRACT | The deliverable D4.15 *Infrastructure Monitoring* is a Technical Specification document that contains an in-depth explanation of the updates made to the i4Q Infrastructure Monitoring solution during the second phase of its implementation. This deliverable provides a comprehensive overview of the strategic planning that was utilized in order to satisfy the remaining pilot requirements. The procedure for solving each pilot task is described in great detail, beginning with an in-depth analysis of the data that is currently at hand, followed by an examination of the approaches that were utilized in the process of resolving the task itself, and concluding with a presentation of the evaluation results that were derived from the utilization of developed method. The conclusions drawn from the experiments | | |

| | |
|---|---|
| | carried out on the industrial pilot datasets provide credence to the efficiency of the Infrastructure Monitoring framework. |

## Document History

| VERSION | ISSUE DATE | STAGE | DESCRIPTION | CONTRIBUTOR |
|---|---|---|---|---|
| 0.1 | 7-Nov-2022 | ToC | First Version of Table of Contents | CERTH |
| 0.2 | 17-Nov-2022 | ToC | Second Version of Table of Contents send to the consortium | CERTH |
| 0.3 | 28-Nov-2022 | Draft | First Draft of the Deliverable to be sent for internal review | CERTH |
| 0.4 | 02-Dec-2022 | Internal Review | Internal review | CESI, DIN |
| 0.5 | 8-Dec-2022 | Draft | Address comments from internal reviewers | CERTH |
| 1.0 | 30-Dec-2022 | Final Document | Final quality check and issue of final document | CERTH |
| | | | | |

## Disclaimer



## Copyright message

# TABLE OF CONTENTS



# LIST OF FIGURES

# LIST OF TABLES

# ABBREVIATIONS/ACRONYMS

| | |
|---|---|
| **AI** | Artificial Intelligence |
| **CNC** | Computer Numerical Control |
| **DNN** | Deep Neural Network |
| **EOL** | End-Of-Line |
| **FFT** | Fast Fourier Transformation |
| **IM** | Infrastructure Monitoring |
| **LGBM** | Light Gradient Boosting Machine |
| **ML** | Machine Learning |
| **PDF** | Portable Document Format |
| **RF** | Random Forest |
| **UI** | User Interface |

# Executive summary

Deliverable **D4.15** *Infrastructure Monitoring v2* is an updated version of **D4.7** that demonstrates the technical advancements of the **i4Q Infrastructure Monitoring solution (i4Q$^{IM}$)**. Specifically, it is a Technical Specification document that showcases a detailed explanation and analysis of the **i4Q$^{IM}$** algorithms established during the second phase of the solution's implementation. First, a summary of the pilot requirements that have been met is shown, followed by a description of the algorithms delivered. The structure of the implementation presentations consists of:

- A description of the pilot tasks and the relevant data.
- A comprehensive description of the approaches used to drive the algorithm's development.
- Quantitative results that summarize the conducted experiments and the metrics involved in the evaluation of the suggested technique.
- Qualitative results that present informative visualisations of the algorithm's performance.

In addition, an introduction of the user interface that will host CERTH's solutions is provided, detailing the technologies utilized and providing a first look at the data visualization and user interaction capabilities of the platform.

This document **i4Q** D4.15 v2 is an update of v1 of D4.7, for this reason it contains information of the 1st version together with the updates developed in this 2nd version.

# Document structure

**Section 1:** Contains the **i4Q Infrastructure Monitoring solution's** technical specifications, including an overview and architectural diagram. It is aimed towards software engineers.

**Section 2:** Describes the updates introduced in the second implementation stage of the **i4Q Infrastructure Monitoring,** providing details on the process of developing the i4Q$^{IM}$ methods and the design of a user interface.

**Section 3:** Provides the conclusions.

**APPENDIX I:** Provides the PDF version of the **i4Q Infrastructure Monitoring solution** web documentation, which can be accessed online at: http://i4q.upv.es/15_i4Q_IM/index.html

# 1. Technical Specifications

## 1.1 Overview

The i4Q$^{IM}$ solution is assigned to the Platform Tier subcomponent Monitor and Diagnostics (Deliverable 2.7). The i4Q$^{IM}$ monitors the health of workloads and the associated processes and generates alerts when a hazardous event is anticipated to be present. AI and ML models are deployed, which are trained and evaluated with historical industrial sensor data. It detects and predicts impending detrimental issues and provides alerts together with the appropriate parameter settings of the entries that are expected to be faulty, so that machine operators or other i4Q solutions can begin reconfiguring in time to avert a total production line stoppage. By analyzing sensor data and utilizing them to further improve the performance of AI and ML models, the i4Q$^{IM}$ solution aims to enable fully autonomous operations across many production lines and processes.

## 1.2 Architecture Diagram

This solution's analytical processes and AI models bridge the i4Q$^{IM}$ to the i4Q Reference Architecture's Platform and Edge Tiers. The advantages of the i4Q$^{IM}$ solution in relation to the aforementioned mapping are as follows:
Strengths:

- **Platform Tier:** The i4Q$^{IM}$ mapping to "Data Brokering and Storage" gives this solution access to a vast array of industrial data. Rich data sources permit the development, fine tuning, and generalization of the predictive models, hence boosting the alerting system's reliability. The mapping between i4Q$^{IM}$ and "Models Management and Services" provides real-time monitoring of other i4Q analytical solutions. The meta-analysis of their analytical results and final outputs enables exceptionally accurate problem detection performances, resulting in effective monitoring of the whole infrastructure. The i4Q$^{IM}$ mapping to the "Monitor and Diagnostics" sub-component enables this solution to operate independently, alerting other analytical i4Q solutions or human operators to take the necessary corrective measures when a machine or process fault is identified. The notion of the predictive alerting system may dramatically reduce faults, avoid frequent production line shutdowns, and prevent costly repairs caused by permanent failures.
- **Edge Tier:** The i4Q$^{IM}$ mapping to the "Data Collecting" sub-component enables the solution to collect industrial data directly from the sensors deployed in the production lines, allowing for real-time problem identification while it operates as an independently service. The i4Q$^{IM}$ mapping to the "Data Management" sub-component guarantees that the solution analyses multi-source manufacturing data (i4Q$^{DR}$ data, i4Q$^{DIT}$ data, sensor data) quickly, resulting in optimal fine tuning of the ML models. The combination of feature engineering and feature importance can disclose crucial information regarding parameter settings associated with the presence of dangerous events.

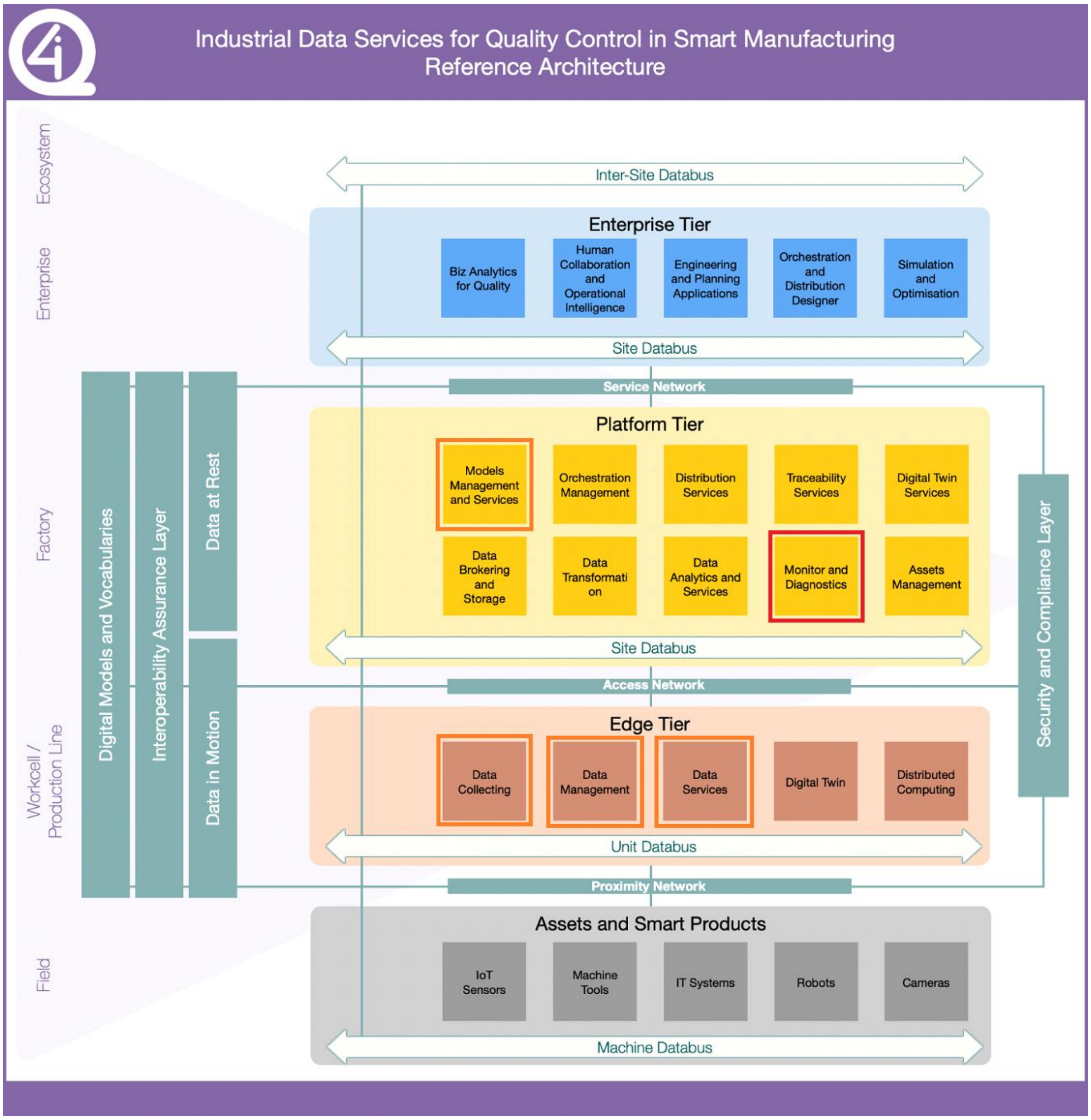


**Figure 1.** i4Q Reference Architecture mapping with i4Q^IM

# 2. Implementation Status

## 2.1 Current implementation

Following the development state of the i4Q$^{IM}$ as outlined in D4.7, the implementation phase of the solution proceeds to fulfil the remaining pilot requirements. The primary objective of the introduced implementations is to develop effective problem detection and proactive warning techniques that could generalized and be adapted to different use-cases. The pilot requirements/use-cases addressed in the document are the following:

- **BIESSE:** The definition of algorithms, in time and frequency domain, that must be performed to predict a component degradation **[BP1&2_PC2r6]** is directly connected to the development of a ML classifier to capable of evaluating whether a CNC tool is worn or not. The i4Q$^{IM}$ solution is called upon to carry out the component degradation detection task by introducing ML algorithms and generating alerts.
- **WHIRLPOOL:** The system shall generate an alert for non-conformity situations to perform Threshold and Importance analyser requirement **[BP01_PC3r2.3]** relates to the machine learning technologies that will be utilized to predict an upcoming manufacturing line issue. The i4Q$^{IM}$ is responsible to detect if a problem is present based on the quality of the manufactured products by employing ML based algorithms and finally warn the operators with alert generations.
- **FACTOR:** The detection of machine failures, a task that is not directly mapped to a specific pilot requirement, is addressed by ML techniques designed to predict impending alarms that are indicative of a specific fault, in an attempt to eliminate permanent damages to the machine. The i4Q$^{IM}$ approaches this task by utilizing AI models and analysing important machine parameters.

### 2.1.1 BIESSE – Component Degradation

#### *2.1.1.1 Data & task description*

The dataset provided by BIESSE is related to the tool wear of the CNC machines containing 10,025 samples of worn and unworn cases along with the associated readings of the machine sensors. The objective of the i4Q$^{IM}$ was to effectively detect the degradation of the tool and provide an alert to the operators to inform them for an imminent failure in order to take action.

#### *2.1.1.2 Methodology*

Since the BIESSE's use case was comparable to the FIDIA's component degradation detection task, the methodology used for its resolution was similar. Therefore, the optimally fine-tuned LGBM model described in D4.7 that was implemented for FIDIA's component degradation case, was also applied on BIESSE's tool wear dataset. A 10-fold cross validation was applied to construct a training set including 8,020 samples (80% of the initial dataset) and a test set including 2,005 samples (20% of the initial dataset). The parameters setting was identical to the ones used in FIDIA's case. The tool wear detection accuracy (test set) was **99.8%**. The careful designing of

LGBM which was conducted by taking into consideration a plethora of parameters and conducting a series of experiments in order to ensure its generalization abilities, has been proven successful. An impressive performance has been achieved without changing the core of the classifier (parameters setting, tree structures, etc.). Thus, the component degradation/tool wear detection framework can be easily and successfully applied on similar use cases, or even on different classification tasks by applying fine-tuning.

#### *2.1.1.3 Qualitative results*

Below are some representations of the qualitative findings of the framework for component deterioration detection applied to the FIDIA use case. The following bar chart depicts the feature importance utilized by LGBM. As shown, the Y real position was the most influential feature to contribute towards the model's predictions.

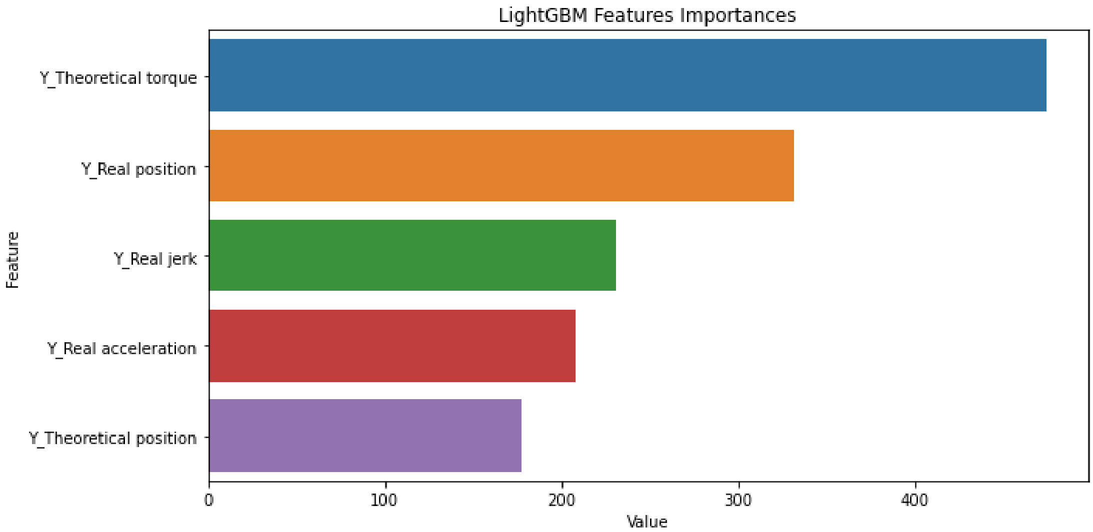


**Figure 2**. LGBM feature importance visualization

### 2.1.2 Whirlpool – Nonconformity situations

#### *2.1.2.1 Data & task description*

The dataset provided by Whirlpool is related to the end of line testing (EOL) performed on the products upon the completion of the manufacturing process. Several tests are performed on the produced washing machines in order to determine the conformity of their quality. The EOL dataset is comprised of 3,516,957 records, the majority of which are instances of good quality products. Specifically, the number of products that have passed the quality inspection is 3,413,169 (97% of the total amount of available samples) whereas the defective ones are only 103,788 (3% of the total amount of available samples). Therefore, it was apparent that the class imbalance governing the data would be a major factor that should be consider in the problem-solving process.

The objective of the i4Q$^{IM}$ solutions is to effectively detect manufacturing line issues based on the occurrence of defective products. In those nonconformity situations, the solution will generate alerts to inform the machine operators that reconfiguration actions should be taken.

#### 2.1.2.2 Methodology

To accomplish the detection of faulty products, several ML algorithms were tested, to narrow down to the best approach. To proceed with the ML experimentations, the dataset was first split into a training and a testing set. The training set includes the majority of the available data (80%), and it is used to train the ML model by learning the underlying patterns of the data. In contrast, the test set is made of the remaining data (20%) and is used to offer an unbiased evaluation of the model's performance once the training phase is complete.

As noted in the description of the pilot dataset there is a plethora of data samples with a severe presence of class imbalance. As a result, class imbalance techniques were incorporated and utilized in the training pipeline of the classification algorithms to ensure their proper training and ultimately obtain ML models that generalize effectively for all possible classes. Tomek Links was the first method utilized to remedy the existing imbalance. It is an under-sampling technique that eliminates some instances of the majority class. Specifically, it discards the majority class samples that are near to the minority class, hence successfully aiding in the separation of the two classes while marginally lowering the sample difference. The second method of addressing the imbalance issue was the utilization of a random under-sampler. This technique is does not follow a specific strategy in the elimination process of the majority class instances. Its goal is to randomly select a subset of the majority class samples to decrease the magnitude of imbalance. Finally, a cost-sensitive strategy was employed to each classifier tested, to punish the misclassification of minority class samples more severely and thus assisting the model to achieve better generalization.

The ML algorithms tested on the use-cases were a Decisions Tree, a Random Forest, a DNN and a LightGBM classifier. The Decision Tree and the Random Forest classifiers could not handle the task as they were running into performance issues due to the sheer volume of data. On the contrary, the DNN and the LGBM classifiers are capable of exploiting larger datasets and due to their incremental learning capabilities allow for model updates when additional data are made available. In the following subsection, the outcome of the first training procedure to establish the optimal algorithm and the fine-tuning process to optimize the chosen model are discussed in detail.

#### 2.1.2.3 Quantitative results

The training process for the LGBM classifiers was conducted within a 10-fold cross-validation. Cross-validation enables the training of generalized models since it prevents the introduction of bias towards a particular data subset. On the other hand, the DNN is comprised of 5 dense layers with 'Adam' optimizer and binary cross-entropy as its loss function. The training of DNN was conducted within 100 epochs. Balanced accuracy was chosen as the metric for evaluating the two algorithms as it considers the presence of imbalance. The following table summarizes the performance comparison of the classifiers within the test set.

| Classifier | Balanced Accuracy |
|---|---|
| DNN | 81.1% |
| **LightGBM** | **99.7%** |

**Table 1.** Performance comparison between DNN and LightGBM

It is clear that the LightGBM outclasses the DNN approach in this particular use case. This performance, however, was not the result of the LGBM using standard settings. Multiple hyper-parameter value combinations have been examined to optimize the performance of the model. This was achieved by utilizing grid-search, which allows testing of different model parameter configurations. The following are the parameters selected for the model optimization and the values tested:

- **Learning rate = [0.01, 0.05, 0.1]**, determining the step size for each loss function iteration.
- **Number of estimators = [100, 160, 240]**, specifying the number of boosted trees.
- **Number of leaves = [16, 32, 64]**, defining the max number of tree leaves.
- **Max depth = [8, 16, unlimited]**, determining the max depth of each boosted tree.
- **Class weight = [{0:1, 1:1}, {0:1, 1:2}, {0:1, 1:3}]**, defining the relative weight of each class.

In the following table, just a subset of grid-search trials is provided. Specifically, these are the five situations with the lowest and highest performance:

| Parameter Settings | Balanced Accuracy |
|---|---|
| ‘class_weight’: {0:1, 1:3}, ‘learning_rate’: 0.01, ‘max_depth’: 8, ‘n_estimators’: 100, ‘num_leaves’: 16 | 54.94% |
| ‘class_weight’: {0:1, 1:3}, ‘learning_rate’: 0.01, ‘max_depth’: 16, ‘n_estimators’: 100, ‘num_leaves’: 16 | 55.59% |
| ‘class_weight’: {0:1, 1:3}, ‘learning_rate’: 0.01, ‘max_depth’: -1, ‘n_estimators’: 100, ‘num_leaves’: 16 | 55.59% |
| ‘class_weight’: {0:1, 1:3}, ‘learning_rate’: 0.01, ‘max_depth’: 8, ‘n_estimators’: 100, ‘num_leaves’: 32 | 58.13% |
| ‘class_weight’: {0:1, 1:3}, ‘learning_rate’: 0.01, ‘max_depth’: 8, ‘n_estimators’: 100, ‘num_leaves’: 64 | 59.60% |
| ‘class_weight’: {0:1, 1:3}, ‘learning_rate’: 0.01, ‘max_depth’: -1, ‘n_estimators’: 240, ‘num_leaves’: 64 | 99.57% |
| ‘class_weight’: {0:1, 1:2}, ‘learning_rate’: 0.01, ‘max_depth’: -1, ‘n_estimators’: 240, ‘num_leaves’: 64 | 99.69% |
| ‘class_weight’: {0:1, 1:2}, ‘learning_rate’: 0.01, ‘max_depth’: 16, ‘n_estimators’: 240, ‘num_leaves’: 64 | 99.69% |
| ‘class_weight’: {0:1, 1:1}, ‘learning_rate’: 0.01, ‘max_depth’: 16, ‘n_estimators’: 240, ‘num_leaves’: 64 | 99.72% |
| ‘class_weight’: {0:1, 1:1}, ‘learning_rate’: 0.01, ‘max_depth’: -1, ‘n_estimators’: 240, ‘num_leaves’: 64 | **99.73%** |

**Table 2.** LightGBM fine-tuning

### 2.1.2.4 *Qualitative results*

In this section, qualitative results produced by the implemented algorithm are presented in the following visualizations. **Figure 3** depicts the data features that contribute the most towards the predictions of the LightGBM classifier.

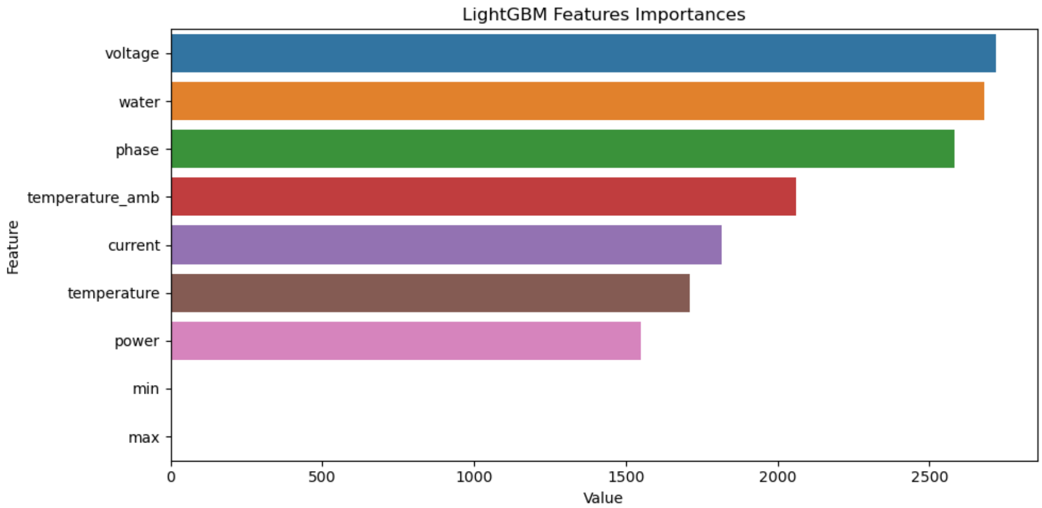


**Figure 3**. LightGBM feature importance visualization

Additionally, **Figure 4** provides a visual representation of the LightGBM prediction performance on the test set.

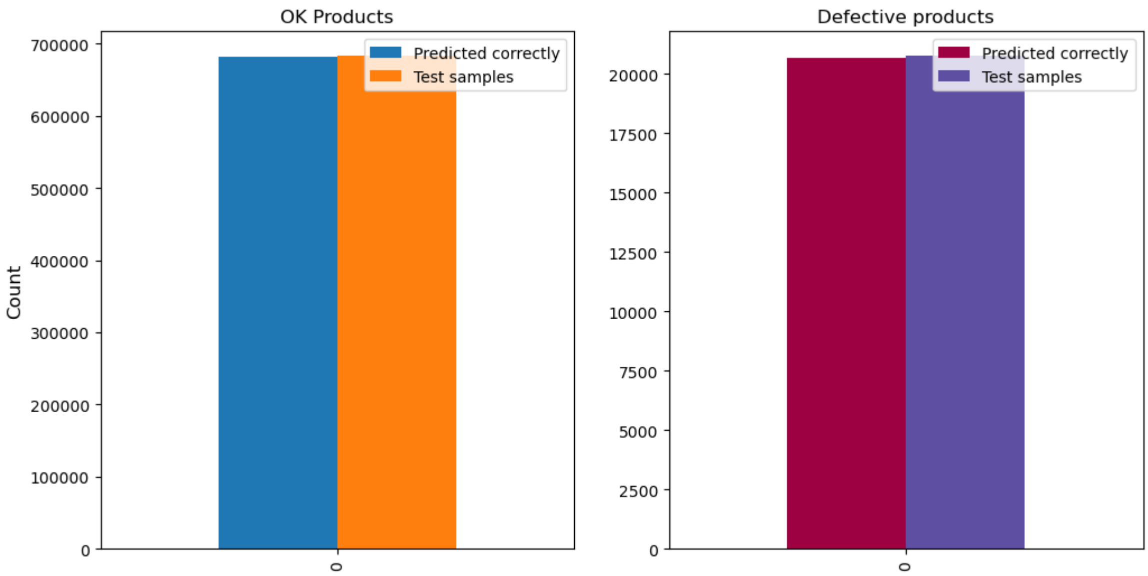


**Figure 4.** LightGBM prediction performance in the test set

### 2.1.3 FACTOR – Machine failure detection

#### *2.1.3.1 Dataset & task description*

The Factor dataset consists of a csv file that contains 37,440 samples of readings from various temperature sensors installed in a cutting tool machine. Each sample is also accompanied by an alarm label that indicates a specific failure of the machine. If the alarm is anything other than 0 then it signifies that the related sample refers to a machine failure. The objective of the i4Q$^{IM}$ solution is to implement an ML algorithm capable of predicting the occurrence of these machine failures.

#### *2.1.3.2 Development issues.*

First of all, several of the samples included in the dataset contained field with missing values. After removing the corresponding samples, there were 31,547 records remained in the dataset. Next, since the goal of the task was to detect whether or not the machine presents a failure, the problem is reduced to a binary classification. Thus, the alarms that indicate a problem were grouped into a single 'failure' class. However, a major problem governing the data was the severe imbalance of samples between the two classes. The instances of machine failure are extremely little to the point where it is not feasible to implement a classification algorithm. Specifically, the total number of machine failure records is only 33, which is the 0.1% of the available data. As a consequence, due to the insufficient number of samples the only option was to test alternative approaches, such as one-class classification techniques that are used for outlier detection. However, even this approach was not able to produce promising results mainly attributed to the absence of data.

## 2.2 User Interface

A front-end mockup of the User Interface (UI) that will host CERTH's i4Q solutions has been developed.

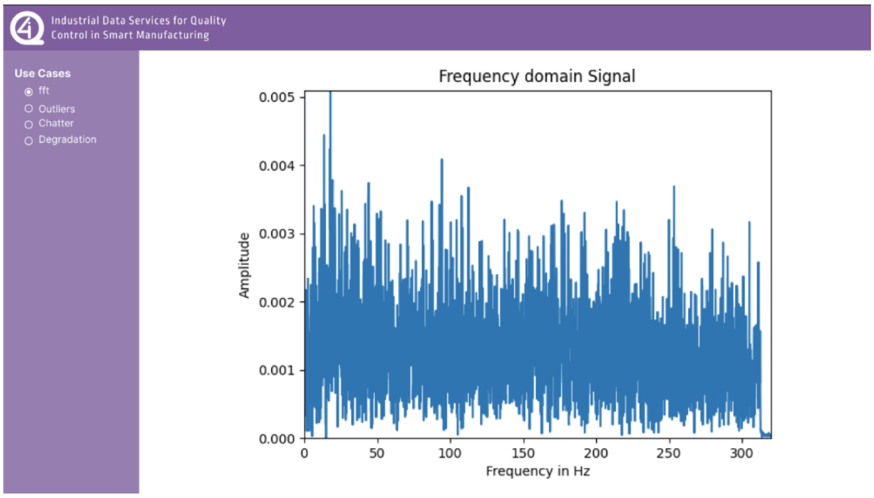


**Figure 5.** User Interface mock-up

The designed UI consists of three elements: **a)** the header, **b)** the side menu, and **c)** the main section holding the chart.

The header includes the i4Q project's logo and name ("Industrial Data Services for Quality Control in Smart Manufacturing"). The user can pick one of the four use cases from the menu on the left side of the interface: FFT, Outliers, Chatter, or Degradation. When a user selects an option, the related graph is displayed in the main section of the interface. It is also possible to zoom in and out of the chart in order to zero in on a particular portion.

The data used to generate the graphs are static and kept in a CSV file. When a use case is selected from the menu, the server reads the corresponding CSV file and delivers the data to the front-end, where they are formatted and sent to the chart for display.

The technologies utilized for the implementation of the UI are HTML5, CSS3 and JavaScript. Chart.js is a JavaScript library that was used to generate the charts and the chartjs-plugin-zoom plugin enabled the implementation of the zoom functionality.

## 2.3 History

| Version | Release date | New features |
|---|---|---|
| V0.7.0 | 25/10/2022 | ML algorithms implementation and evaluation to cover the remaining pilot requirements |
| V0.7.5 | 20/11/2022 | Front-end mock-up for the User Interface |
| V1.0 | 30/12/2022 | Final version |

**Table 3.** History

# 3. Conclusions

Deliverable D4.15 Rapid Quality Diagnosis is a technical specification document that provides a full overview of the updates made in the second stage of the i4Q[IM] implementation phase. It provides complete information on the design process of the analytical algorithms intended to meet the remaining pilot requirements, detailing every development step.

The CNC tool wear/component degradation detection algorithm that was developed for FIDIA, as described in D4.7, was proven to be extremely effective in similar scenarios such as BIESSE's case. The LGBM classifier after being fine-tuned on BIESSE's data and without further complex modifications, was able to predict component defects with an accuracy of 99.8%. This indicates that the i4Q[IM] has successfully generalized the implementation of its CNC tool wear/component degradation detection framework.

The LGBM classifier also performed exceptionally in WHIRLPOOL's use case. The incorporation of class imbalance methods into the training pipeline of the ML model, such as random under-sampling, Tomek links, and cost-sensitive class weighting, significantly contributed to the model's performance. Through fine-tuning, the model obtained a balanced accuracy of 99.7% in the detection of non-conformity product quality that may indicate a manufacturing line issue.

In the case of the FACTOR the lack of sufficient data was proved to be a deterrent to the implementation of an algorithm that could effectively detect machine failure. It is apparent, that the provision of additional data is necessary.

Finally, an introduction to the user interface was presented, detailing the utilized technologies and providing a sneak peek at its data visualization and user interaction capabilities.

# Appendix I

**i4Q Infrastructure Monitoring solution (i4Q$^{IM}$)** web documentation can be accessed online at: http://i4q.upv.es/15_i4Q_IM/index.html