ZDMP: Zero Defects Manufacturing Platform

WP8: Product Quality Assurance

EU ID: D107: Pre-Production: Product Quality Prediction - Vs: 1.0.0A

ZDMP ID: D8.2a

Deliverable Lead and Editor: Mauro Fabrizioli, VSYS

Contributing Partners: CET, ITI, VSYS

Date: 2020-06

Dissemination: Public

Status: EU Approved

Abstract

The deliverables for this task, and all WP5-8 tasks, are software and are of EU type “OTHER”. The software and accompanying material (eg description, instructions) is available on the ZDMP software repository which is updated dynamically. However, for EU formal reporting purposes, this brief cover document provides a formalised pointer to the downloadable software and related content. This deliverable should read in conjunction with the D006-D020 deliverables which document the software process/status for each WP/Task. This deliverable represents the status as at M18 with further living editions at M18 and M48

Grant Agreement: 825631

Document Status

Deliverable Lead	Mauro Fabrizioli, VSYS
Internal Reviewer 1	Oscar Salgado, IKER
Internal Reviewer 2	Jindrich Svoboda, FORM
Internal Reviewer 3	Stuart Campbell, ICE
Type	Deliverable
Work Package	WP8: Product Quality Assurance
ID	D107: Pre-Production: Product Quality Prediction
Due Date	2020-06
Delivery Date	2020-06
Status	EU Approved

History

See Annex A.

Status

This deliverable is subject to final acceptance by the European Commission.

Further Information

www.zdmp.eu and mailto:info@zdmp.eu

Disclaimer

Project Partners:

UNINOVA Instituto de Desenvolvimento de Novas Tecnologias

ICE Information Catalyst Services Data Software

Ford Go Further

MONDRAGON ASSEMBLY

HSD MECHATRONICS

FIDIA

FORMPLAST SVOBODA

CONSULGAL CONSULTORES DE ENGENHARIA E GESTÃO, S.A.

PT MILLS

FLEXEFELINA STEEL TUBES

cei by ZIPOR

Continental

AlfaTest Your automation and test partner

DIN

software AG

SIMAVI Software Imagination & Vision

ceteck

VIDEOSYSTEMS Technologies for a new world

ASCORA Consulting - Development - Research

PROFACTOR

algowATT GREEN TECH SOLUTIONS

Etxetar

AL ANTÓNIO LONGARITO the art of stone

rooter

ITI INVESTIGATE TO INNOVATE

UNIVERSITAT POLITÈCNICA DE VALÈNCIA

ikerlan MEMBER OF BASQUE RESEARCH & TECHNOLOGY ALLIANCE

iT innovation

MARTINREA

Tampere University

Executive Summary

The main objective of “WP8: Product Quality Assurance” is to ensure the quality of the product along the value chain of the manufacturing process by developing zero defect manufacturing (ZDM) applications based on digital models of manufacturing assets and manufacturing processes. The deliverables of this work package and the WP1 Management work package are divided into software packages and document/reports. In terms of reporting:

- **Process/Status:** Report D107 Pre-Production: Product Quality Prediction of WP8 Product Quality Assurance, as identified in the DOA, focuses on the process/status of the work accomplished in Task T8.2
- **Software:** All WP8 software deliverables of T8.1-T8.4 (type “OTHER”) are available in the ZDMP public repository with access details and install instructions further described in this report which is a ‘current’ extract of the repository

“WP8: Product Quality Assurance” consists of: Modelling, Prediction, Inspection, and Supervision. The tasks of WP8 are the following:

- T8.1 - Characterization and Modelling / Digital Twin
- T8.2 - Pre-Production: Product Quality Prediction / Product Assurance Runtime - Quality Prediction
- T8.3 - Production: Non-Destructive Product Inspection / Non-Destructive Inspection
- T8.4 - Production: Supervision / Product Assurance Runtime – Quality Supervision

This deliverable represents Task T8.2 Pre-Production: Product Quality Prediction which in turn is composed of the following components:

- Product Assurance Run-Time – Product Quality Prediction

As reported in the architecture deliverable the purpose of these components is: “To assure product quality. Specifically, ZDMP must provide product-oriented services that can guarantee a zero-defect scenario along the whole manufacturing process of the product: Modelling, Pre-Production, Production, and Supervision. This component implements a set of machine learning (ML) models to support this goal for the Pre-Production and Supervision phases”.

Each of the components is structured into the following sections:

- General Description
- Architecture Diagram
- Features
- Requirements
- Installation
- How to Use
- Functional Requirements Implementation Status (M18)

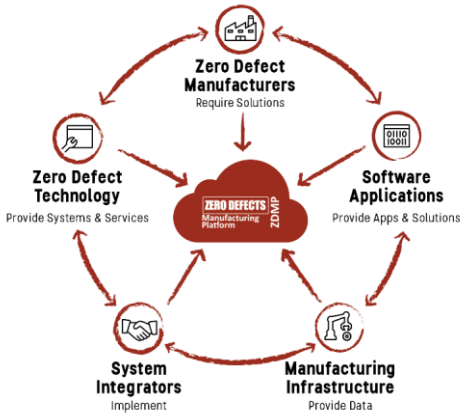
This report covers the period from the project start until M18 with most activity in the M13-M18 period. Further formal deliverables are due M30 and M48 as well as an informal iteration at 24.

Table of Contents

0 Introduction

Due to the cover nature of this deliverable; this introduction is presented in short form only. For further information please consults D006 - Technical Management: Overview Report.

0.1 ZDMP Project Overview



ZDMP – Zero Defects Manufacturing Platform – is a project funded by the H2020 Framework Programme of the European Commission under Grant Agreement 825631 and conducted from January 2019 until December 2022. It engages 30 partners (Users, Technology Providers, Consultants and Research Institutes) from 11 countries with a total budget of circa 16.2M€. Further information can be found at www.zdmp.eu.

ZDMP aims at providing such an extendable platform for supporting factories with a high interoperability level, to cope with the concept of connected factories to reach the goal of zero-defect production. For this, the platform provides the tools to allow following each step of production, using data acquisition to automatically determine the functioning of each step regarding the quality of the process and product.

0.2 Deliverable Purpose and Scope

The deliverables for this task, and all WP5-8 tasks, are software and are of EU type “OTHER”. The software and accompanying material (eg description, instructions) is available on the ZDMP software repository which is updated dynamically. However, for EU formal reporting purposes, this brief cover document provides a formalised pointer to the downloadable software and related content. This deliverable should read in conjunction with the D006-D020 deliverables which document the software process/status for each WP/Task. This deliverable represents the status as at M18 with further living editions at M18 and M48. Specifically, the DOA states the following regarding this Deliverable:

T8.2	**Pre-Production: Product Quality Prediction**			**VSYS**	**Six Monthly**
D107 D108 D109	Pre-Production: Product Quality Prediction	OTHER (Prototype)	PU	18, (24), 30, 48, Reporting via T1.4.x Series	RDI3-6
The aim of this task is to predict the quality of the product and anticipate possible defects arising during its manufacturing process. The task will use the digital model of the processes developed in T7.1 and will apply the AI tools and data analytic services provided by the ZDMP Platform, with the aim of developing statistical quality control techniques to infer relationships between the different parameters along the product value chain and the final quality of the product. Moreover, critical parts will be prototyped in laboratory tests to validate the digital models in a sample and trial basis, and to outline the most significant variables or quantities to be observed during factory production. The output of this task will support task T7.1 on the decision-making during the early stages of manufacturing, including tuning and adjusting of parameters, and the application of corrective measures.					

0.3 Target Audience

The primary target audience for this document are the partners and WPs of the project, as well as the EU and reviewers.

0.4 Deliverable Context

The deliverable context is as per Section 0.2:

Primary Preceding documents:

- **D006: Technical Management Overview Report:** Represents the general software status of the project including information on commits and WP5-8 Risks and mitigations
- **D018: Technical Management: WP8 Report:** Represents the process/status and future actions of this work package, including this task. It also includes related KPIs and their status
- **D055: Technical Specification and Update:** Describes the different APIs of the components

0.5 Document Structure

This deliverable is broken down into the following sections:

- **Section 1: Component: Product Assurance Run-time – Product Quality Prediction**

0.6 Document Status

This document is listed in the Description of Action as “public” since it represents the open nature of the project’s software deliverables.

0.7 Document Dependencies

- None

0.8 Glossary and Abbreviations

A definition of common terms related to ZDMP, as well as a list of abbreviations, is available at http://www.zdmp.eu/glossary.

0.9 External Annexes and Supporting Documents

- See the ‘Resources’ grid within the General Description Section of each component

0.10 Reading Notes

- None

0.11 Document Updates

- This is the first version of this document

1 Component: Product Assurance Run-time – Product Quality Prediction

1.1 General Description

The Product Assurance Runtime objective is to perform actions to ensure the quality and suitability of a product, based on the data obtained in the manufacturing process, or in the inspection of the products once they are finished. Internally, the Product Assurance Runtime is divided in two different tasks:

- Product Quality Prediction, related to Task T8.2 Pre-Production: Product Quality Prediction
- Supervision, related to Task T8.4 - Production: Supervision

Both tasks are based on Artificial Intelligence (AI) and Big Data techniques. However, the approach vary as the desired results are different, and the techniques, or the way they are applied, will also be different.

The next section is a generic introductory description of the component as a whole, but since this deliverable represents Task T8.2 – Pre-Production: Product Quality Prediction, the following sections starting from Section 1.2.2 focus on the description of its specific features, requirements and use.

Resource	Location
Source Code	Link
Latest Release (v1.0.0)	Download
Open API Spec	Link
Video	Coming soon

The date of generation of this component content is: 2020-06-29

1.2 Architecture Diagram

The following diagram shows the position of this component in the ZDMP architecture.

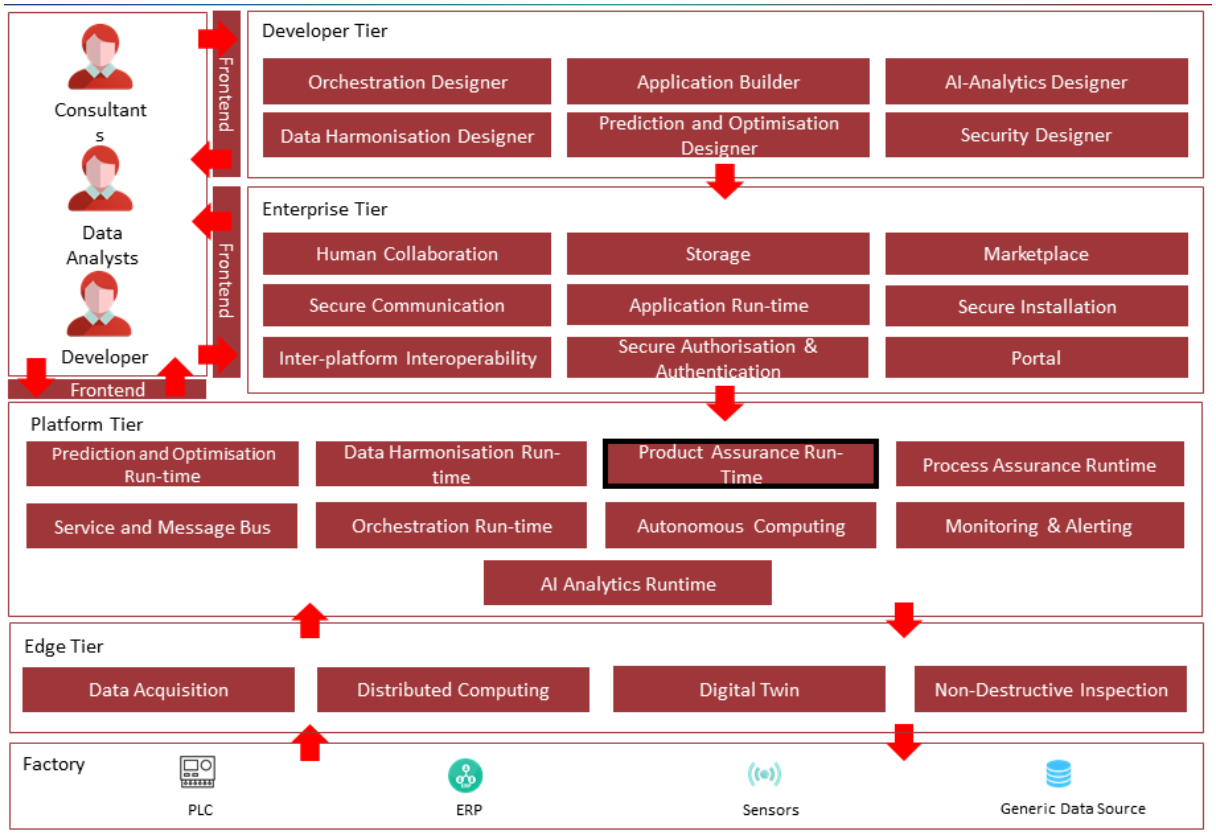


Figure 1: Position of Component in ZDMP Architecture

1.2.1 Product Assurance Run-time Description

The following figure shows an example of use in the automotive use case and Product Assurance Runtime's main interactions with other components:

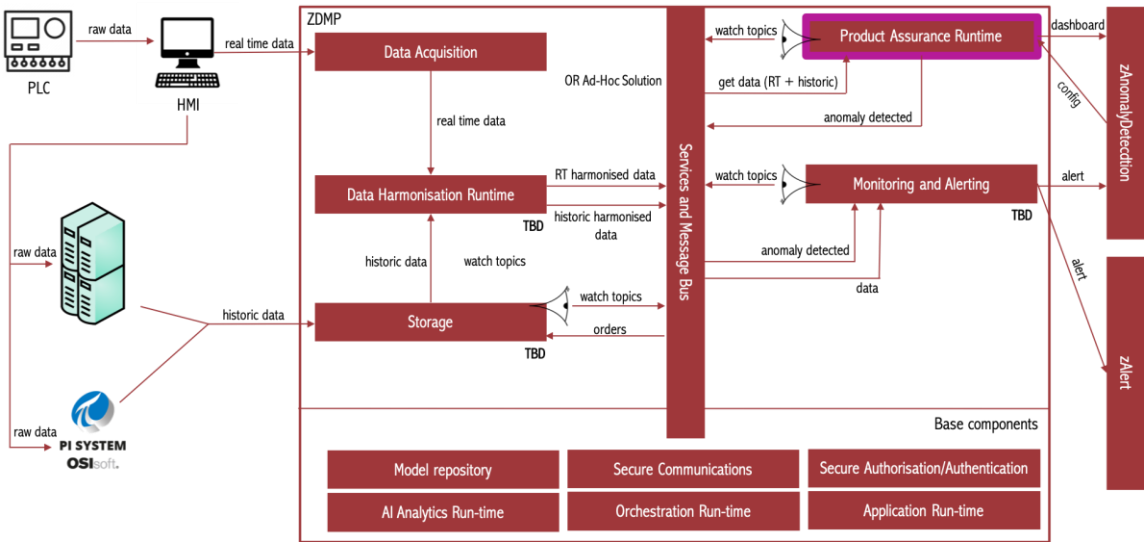


Figure 2. Main interactions of Product Assurance Runtime in automotive pilot.

To illustrate the above figure, an example of a basic flow is described below:

- The sensors placed in the industry manufacturing process send data to an HMI (Human Machine Interface)
- The HMI can perform small 'ad-hoc' specific transformations of the data if it is needed, and then forwards the data to one of the following:
    - An external database in which is stored the historic raw data, ideally a Big Data database
    - ZDMP's Data Acquisition Component, as real time data
- The Data Acquisition component retrieves the real time data and forwards it to the Data Harmonisation component, if needed
- Once the data is harmonised, it is pushed to a specific topic to the ZDMP's Service and Message Bus component. All the components interested in that topic will be watching and will receive the data as soon as is available
- Product Assurance Runtime works in two different modes: Training and prediction
- In training mode:
    - Product Assurance Runtime asks for a set of historic data to the ZDMP's Storage component through an operational message in a specific topic of the Message and Service Bus component. The following message is an example, the final syntax is not complete

```
{
  "operation":  "storage.retrieve.data",
  "topic":      "product.quality.assurance.historic.data",
  "uniqueId":   "ae68ac7f-98b3-4cdd-ba66-c83d93bf5206",
  "payload": {
    "sensors": [
      "sensor-identifier-1",
      "sensor-identifier-2",
      "sensor-identifier-3",
    ],
    "startDate":  "2020-01-22 11:00:00am",
    "endDate":    "2020-01-30 11:00:00am"
  }
}
```

    - ZDMP's Storage component will be watching this topic and should execute the specified operation. In this case, retrieve historic data of the sensors listed between the dates specified
    - Once the operation is executed, ZDMP's Storage component sends the results to another topic in the Service and Message Bus, in which Data Harmonisation component is watching to perform some transformations
    - When the Data Harmonisation component finishes their transformations, it pushes data again to another topic in the Service and Message Bus.
    - Product Assurance Runtime will be watching this topic. Thus, the data is retrieved and sent to the selected subsystem, ie Product Quality Prediction or Supervision
    - Once the data is available, a new model is trained and stored, based on that historic data. The result of the training is a Docker Image which contains an executable instance of the Artificial Intelligence model

  - That Docker Image is sent to the AI Analytics Runtime, with a set of metadata in a manifesto (json) file: this image will be available for the prediction phase
- In prediction mode:
  - In this case, Product Assurance Runtime receives a continuum of real time data from the ZDMP's Service and Message Bus. As in the previous phase, Product Assurance Runtime is watching to a specific topic, pulling the data as soon as it is available
  - Depending on the tool selected, the number of tasks executed will vary, but all of them will perform a prediction with the data retrieved above
  - The result of the prediction is pushed to a specific topic of the Message and Service Bus, in which the interested components can retrieve that information and perform their actions, eg: alerting if an anomaly is detected

1.2.2 Product Quality Prediction Description

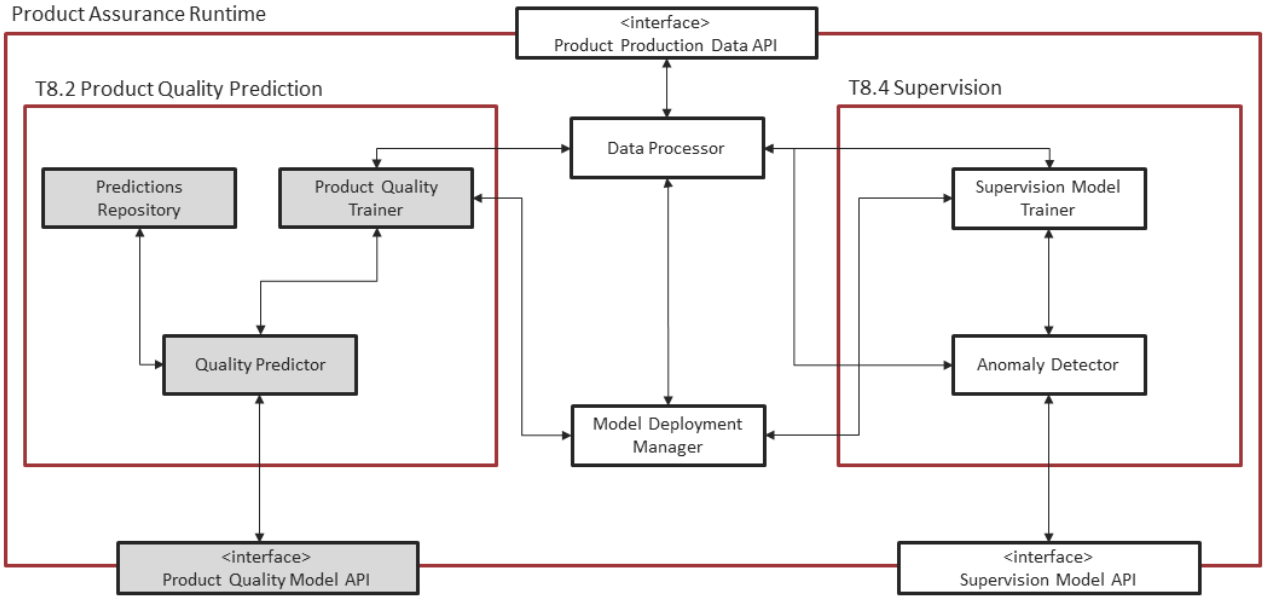


Figure 3. Subsystems architecture of Product Quality Prediction task

The main objective of the Product Quality Prediction task is to provide real time predictions about the product quality variables of a manufacturing process through Machine Learning Models and Big Data techniques.

The main idea of this component is to provide some auto machine learning tools, helping the user to build a trained model for future quality predictions in both classification and regression scenarios. Since many classical models need to be parametrized, these are best called hyperparameters (eg max depth for Random Forest, penalty and loss for Support Vector Machine, iterations and regularization for Linear Ridge, …), this component also provides a Bayesian optimization using Gaussian Processes techniques to optimize the selected model. After the model has been fitted the user can call it to make predictions.

Thus, the Product Quality Prediction task is split into two steps:

- Training
- Inference, ie prediction at run-time

1.3 Features

1.3.1 Training

At this stage all the features needed to optimize and train the model are managed as parameters of train_opt class available in train.py library:

- Data Loader and Pre-processing:
    - Provide functionalities to load and split the dataset where the variable to predict is the last column
    - Automatic relabelling for classification task
    - Opportunity to reduce the number of classes for classification task
    - Generate a new feature matrix consisting of all polynomial combinations of the features with degree less than or equal to the specified degree
    - Standard Normalization of training variables
- Hyperparameters Optimizer:
    - The library provides the Hyperparameters Optimization available for some models (Linear Regression for example does not have hyperparameters to optimize) with respect to a metric selected by the user
    - The algorithm is based on Bayesian optimization using Gaussian Processes; The model is trained on a search space that include Categorical, Real, Integer variables depending on the model parameters
    - At this stage a user-friendly version is provided where the hyperparameters space to optimize has been already selected; in the future a more customized version will be provided for expert users where it will be possible to select which hyperparameters to optimize, the call-backs to stop the optimizer and some other optimizer's parameters
- Models and Metrics. The implemented models to optimize and the available metrics are listed below:
    - Regression task:
        - Models:
            - XGBRegressor
            - LinearRidge
            - LinearRegression
        - Metrics:
            - explained_variance
            - neg_mean_absolute_error
            - neg_mean_square_error
            - r2
    - Classification task:
        - Models:
            - RandomForest
            - LinearSvm

    - Metrics:
        - f1_weighted
        - precision_weighted
        - jaccard_weighted
        - recall_weighted
        - accuracy
- Feature importance and plots:
    - Provide plots for feature importance based on permutation importance algorithm. It could be possible to use this information for further analysis
    - Provide plots of optimizer's steps during training
- Save model:
    - The trained model is stored with onnx extension on a checkpoints folder chosen by the user
    - The whole pipeline (normalization, polynomial features, …) is frozen and stored in the onnx model

1.3.2 Inference

The library *inference.py* allows to load a onnx trained model and make a run-time prediction. Since the model contains the pre-process pipeline there is no need to normalize the data before inference. The input data format must remain the same as for the training step, except for the predict variable's column, eg if training sample was [x_1, …, x_n, y] where y was the variable to predict, only pass only [x_1, …, x_n] to make a prediction.

1.4 System Requirements

1.4.1 Training

This library is implemented using Python 3.6 and the following Python libraries:

Software	Version
numpy	1.18.1
pandas	1.0.1
sklearn	0.21.3
skopt	0.7.3
xgboost	1.0.32
onnx	1.7.0
skl2onnx	1.6.1
onnxmltools	1.6.1
matplotlib	3.2.0
tqdm	4.43.0

1.4.2 Inference

This library is implemented using Python 3.6 and the following Python libraries:

Software	Version
numpy	1.18.1
pandas	1.0.1
onnx	1.7.0
onnxruntime	1.3.0

1.5 Installation

1.5.1 Training

No formal installation is required to run training phase:

- Locate Python code, stored inside “utils_” folder
- Copy “utils_” folder in the folder where is located the code of “train.py” library
- The checkpoints folder is created with the trained model stored inside

```
checkpoints/
└── model.onnx
train.py
utils_/
├── loader.py
├── minimizer.py
├── models.py
└── plot.py
```

1.5.2 Inference

No formal installation is required to run prediction phase:

- Pass the path where is stored the model to the load_model class provided by inference.py library

```
checkpoints/
└── model.onnx
inference.py
```

1.6 How to use

1.6.1 Training

Follow the next steps:

- Import the train_opt class to train the model and hyperparameters

```
from train import train_opt
```

- train_opt class parameters:
  - task possible values:
    - regression (default)
    - classification
  - model (default = None): See the available models in Features section, if no model is provided the algorithm will test all the models available for the task and select the best one
  - metric (default = None): See the available metrics in Features section**.** The default metric is related to the task:
    - If task is regression, then metric should be: “neg_mean_squared_error”
    - If task is classification, then metric should be: “accuracy”
  - ckp_fold (default = “checkpoints”): folder where to save the fitted and optimized model
- Parameters of method fit (self, data, split_rate, n_iter, degree, plot):
  - data: The dataset to be used during training, where the last column is the variable to predict
  - split_rate (default = 0.8): Split dataset into random train and test subsets. The value 0.8 represent the proportion of the dataset to include in the train split.
  - n_iter (default = 20): Number of iterations to optimize the model’s hyperparameters; each iteration provides a fitted model over a subspace of the possible hyperparameters
  - degree (default = None): Build a new dataset with polynomial features of degree=n (int)
  - plot (default = False): Plot permutation importance and optimizer’s steps
- Run train_opt, using Panda’s Data Frames

```
df_train = pd.read_csv("path\file\train.csv")
optimizer = train_opt(task=”regression”, model=”LinearRegression”)
optimizer.fit(data=df_train, degree=2, plot=True)
```

1.6.2 Inference

When the model is trained, it can be used to make predictions following the next steps:

- Import the predictor class and load the onnx model from the checkpoints folder

```
from inference import load_model

model_path = ("checkpoints\model.onnx")
run_time_pred = load_model(model_path)
```

- One sample or more than one sample could be processed. The format must be numpy.array([sample_1, …, sample_N]) where each sample is a numpy array of dimension 1. If data are loaded as panda’s data-frame and to select one sample do as follows:

```
df = pd.read_csv("path\file\data.csv")
```

```
sample = df.iloc[i:i+1, :].to_numpy()
#or
sample = np.array([df.iloc[i, :]])
results = run_time_pred(sample)

#or
samples = df.to_numpy()
results = run_time_pred(samples)
```

where results will be a list containing the predicted values.

1.7 Functional Requirements Implementation Status (M18)

The actual implementation status vis-à-vis the functional requirements implementation at M18 is provided in the annex of the D006 Technical Management Overview Report. This represents the general software status of the project and this WP/Task including information on commits and WP5-8 Risks and mitigations. Below is shown a dummy example for a security component.

Functional requirement	Description	Status	Progress	Comments
T52A013 - Issue New certificates	New client certificates are created. These certificates include the details that permit the identification of the subject (physical device, gateway, or server).	Working	90%	Beta version, requires integration API with security command centre for credentials tokenization

2 Conclusions

This deliverable is the first deliverable in the reporting series for T8.2 Pre-Production: Product Quality Prediction. The deliverables for this task, and all WP5-8 tasks, are software and are of EU type “OTHER”. The software and accompanying material (eg description, instructions) is available on the ZDMP software repository which is updated dynamically. However, for EU formal reporting purposes, this brief cover document provides a formalised pointer to the downloadable software and related content.

This deliverable should read in conjunction with the D006-D020 deliverables which document the software process/status for each WP/Task vs its content. This deliverable represents the status as at M18 with further living editions at M18 and M48 and an informal iteration at M24.

ZERO DEFECTS
Manufacturing
Platform

ZDMP

www.zdmp.eu