D5.1 Industry 5.0 EDGE AI Foundations

Person responsible / Author:	SUITE5
Deliverable N.:	D5.1
Work Package N.:	WP5
Date:	09/10/2023
Project N.:	101092069
Classification:	Public
File name:	Industry 5.0 EDGE AI Foundations
Number of pages:	40

Funded by
the European Union

Status of deliverable

Action	By	Date (dd.mm.yyyy)
Submitted (author(s))	SUITE5	03/10/2023
Responsible (WP Leader)	SUITE5	09/10/2023
Approved by Peer reviewer	LTU	09/10/2023

Revision History

Date (dd.mm.yyyy)	Revision version	Author	Comments
07/06/2023	0.1	SUITE5	Table of Contents
10/06/2023	0.2	SUITE5	Revised Table of Contents
30/08/2023	0.3	SCCH	Input for Sections 3 and 4
01/09/2023	0.4	SUITE5	Additions and comments to all sections
14/09/2023	0.5	HOPU	Input for Section 5
18/09/2023	0.6	SUITE5, SCCH	Revisions of Sections 3 and 4
20/09/2023	0.7	SUITE5	Input for Sections 1, 2 and 6 and comments for other sections
02/10/2023	0.8	SCCH, HOPU	Updates in respective Sections based on comments
03/10/2023	0.9	SUITE5	Final version ready for Peer Review
06/10/2023	0.95	LTU	Peer Reviewed Versions
09/10/2023	1.00	SUITE5	Final deliverable Version

Author(s) contact information

Name	Organisation	E-mail	Tel
Nefeli Bountouni	SUITE5	nefeli@suite5.eu	
Sotiris Koussouris	SUITE5	sotiris@suite5.eu	
Jorge Martinez Gil	SCCH	jorge.martinez-gil@scch.at	+43 50 343 838
J. Eleazar Escudero	HOPU	j.eleazar@libelium.com	

Table of Contents

Figures

Tables

Abbreviations and Acronyms:	
AI	Artificial Intelligence
API	Application Programming Interface
AutoML	Automated Machine Learning
CI/CD	Continuous Integration and Continuous Deployment
CI	Collaborative Intelligence
HITL	Human in the Loop
HOTL	Human on the Loop
IoT	Internet of Things
ML	Machine Learning
MLOps	Machine Learning Operations
NLP	Natural Language Processing

1. Executive Summary

Deliverable D5.1 constitutes the first version of the outcomes of tasks T5.1 – "Collaborative Intelligence Platform for Industry 5.0" and T5.2 - "Open Hardware Platform for Embedded AI and AI-at-the-Edge" at [M9] of the project. The current documentation accompanies the initial results and designs of both platforms that are available through the early deployment of the Edge AI reference library, openly available under the URL http://wiki.ai-redgio50.s5labs.eu, and furthermore this deliverable reports on the methodology, research work and specification of the platforms.

Work performed and included in the context of D5.1 has evolved around three axes that ran in parallel, with continuous collaboration and cross-task communication to ensure alignment:

- the first axis evolves around the definition of the AI-ML Ops Lifecycle and the identification of existing solutions facilitating each step of the Lifecycle. The AI-ML Ops Lifecycle is composed of three main phases: Design, Model Development and Operations. Each phase consists of three sub-phases, resulting in a total of nine steps to complete the total cycle. In the context of AI REDGIO 5.0 WP5, the AI MLOps Lifecycle provides a framework for a structured analysis of AI needs and highlights the possible points of enhancement, wherein the WP5 tools can be introduced by any user that wants to design, implement, and deploy an AI model or pipeline. Additionally, the identified existing tools can be valuable resources for AI REDGIO5.0 experiments and external users, to employ to facilitate their own processes – once they are available through the public Edge AI Reference Library.
- Afterwards, we proceeded with the exploration of business needs with regards to cloud-edge AI and collaborative intelligence, through bilateral interviews with the use cases, focusing on AI aspects, including also collaborative intelligence and cloud-edge requirements, while also exploring the maturity of the use cases following the AI-ML Ops lifecycle and identifying the availability of data. This process led to the extraction of an initial set of requirements for the Collaborative Intelligence (CI) Platform and the Open Hardware Platform, to deliver solutions of added value to the experiments that can act as representative use cases from the manufacturing sector.
- The third axis concerns the encapsulation of findings into the actual progress of the platforms and in particular: on the one hand, the design of the AI REDGIO 5.0 Collaborative Intelligence Platform and delivery of a first version of mock-ups illustrating the envisioned features. And secondly, the specifications, the experimentation and development activities on the Open Hardware Platform to ensure the smooth onboarding and execution of the experiments' AI models and pipelines at the edge at a later stage, resulting in a set of clear specifications and indicative examples of use.

The next steps of T5.1 and T5.2 include the development of the CI Platform features and the improvement of the Open Hardware Platform and its services relevant to the execution of AI models, based on the specifications reported in the wiki. The developed solutions will undergo user validation in the context of WP6 for experiments that have envisioned HITL and edge AI in their use cases and technical validation through the integration activities of T5.6.

The fully developed solutions will be delivered in the second and final iteration of this deliverable, in the context of D5.4 [M27], and they will encapsulate the complete set of features, along with adaptations and advancements based on the user feedback and the results of the technical verification activities.

2. Introduction

2.1. Scope of the Deliverable

Deliverable D5.1 – "Industry 5.0 EDGE AI Foundations" encapsulates the results of the activities performed in the context of WP5, and under tasks T5.1 – "Collaborative Intelligence Platform for Industry 5.0" and T5.2 - "Open Hardware Platform for Embedded AI and AI-at-the-Edge" towards the design and advancement of the first version of the AI REDGIO 5.0 Collaborative Intelligence Platform and the Open Hardware Platform respectively. Furthermore, as D5.1 constitutes the first deliverable of WP5, the background work performed here – i.e., the brief landscape analysis on relevant AI paradigms, and the identification of the experiments' AI needs - also set the foundations for the design and development of the rest of the WP5 tools, that aim to work together and deliver at the end an integrated solution for edge AI to the end-users.

The Collaborative Intelligence Platform for Industry 5.0 aims to facilitate human – AI teaming in manufacturing settings, through the implementation of the Human-in-the-Loop paradigm, with the provision of the required human-AI interaction, feedback and adaptation mechanisms that allow hybrid teams maximise benefits from collaboration.

The first version of the CI Platform contains a refined set of requirements and the mockups that illustrated the envisioned functionalities and interactions that enable human – AI teaming. The AI REDGIO 5.0 Open Hardware Platform enables the deployment and execution of AI models on open-source hardware, thus enabling users take advantage of the edge devices' computational power, and leverage on the potential of edge AI.

The first release of the Open Hardware Platform includes a comprehensive list of specifications for users that want to make their AI models compatible with the open hardware, while the work performed included advancements in the platform to support in action the deployment of indicative AI models as proof-of-concept.

The tangible results of this work, including the mockups of the Collaborative intelligence Platform and the specifications and model compilation and deployment aspects of the Open Hardware Platform are available through the early deployment of the Edge AI reference library, openly available under the URL http://wiki.ai-redgio50.s5labs.eu/.

Following the project's iterative approach, both solutions, i.e., the Collaborative Intelligence Platform and the Open Hardware Platform, are planned to be delivered in two iterations, in the context of the deliverable at hand, i.e., D5.1 [M9], and in the upcoming deliverable D5.4 [M27]. The former contains the early results of the relevant tasks and the latter will deliver the fully developed solutions, with features that cover the full set of defined requirements, as well as any enhancements and adaptations based on the lessons learnt from their integration and technical verification in the context of D5.3 [M18] and D5.6 [M33] and the validation activities in the context of WP6.

2.2. Relations to other WPs and Deliverables

Deliverable D5.1 reports on the results of T5.1 and T5.2 until M9, and particularly, it encapsulates the work on needs identification, frameworking, experimentation and the initial mock-ups and results related to the Collaborative Intelligence Platform and the Open Hardware.

For the design of both platforms, inputs were received with regards to business needs expressed through the User Scenarios of WP2 and WP6, and data-related aspects from WP4. Additionally, as WP5 aims to deliver an integrated toolkit of AI solutions for Edge and Embedded AI, the foundational work performed in the context of this deliverable, feeds the other tasks of WP5, namely: the Edge AI Cloud-to-Edge AI Pipeline

Lifecycle Management Platform (T5.3), the Edge AI reference library (T5.4), and the Interoperability with AI-on-demand platform (T5.5). Additionally, as all WP5 tools will be integrated in the context of T5.6, particular attention should be provided for cross-task alignment to ensure tools interoperability.

With regards to requirements engineering, in the context of WP2, WP3 and WP6, D5.1 can be used as input for the concrete specification of edge AI and collaborative intelligence needs that could not be specified in detail beforehand. The Techno Handbooks corresponding to the tools of D5.1 will be updated accordingly to reflect the latest status. Additionally, the Data Solutions of WP4 and the Reference Architecture (Data Spaces, Pipelines etc.) should take into consideration interoperability and compatibility aspects, based on the specification of the CI Platform and the Open Hardware Platform.

Finally, the results of D5.1 will be validated from the user perspective in the context of WP6 activities and will undergo technical verification under T5.6 with the findings and lessons learnt to be included in D5.3 [M18]. These findings and the results of the additional development work in T5.1 and T5.2 will be integrated in the delivery of the second and final version of the Collaborative Intelligence Platform and the Open Hardware Platform, in the context of D5.4 on [M27].

3. Cloud – Edge AI ML Ops Lifecycle

3.1. MLOps Lifecycle

The emergence of Artificial Intelligence (AI) and Machine Learning (ML) has revolutionised multiple industries, ranging from healthcare and finance to manufacturing and transportation. With the fast growth of data availability and the need for real-time decision-making, Cloud-Edge AI ML Operations (AI MLOps) has become a powerful approach, being able to combine cloud computing, edge devices, and advanced ML algorithms.

This section introduces the Cloud-Edge AI ML Ops lifecycle, explaining the stages of deploying, managing, and optimising AI and ML models in this distributed and dynamic environment. Figure 1 illustrates the complete MLOps lifecycle.

As we can see in the next figure, the life cycle is composed of three main phases: Design, Model Development and Operations. Each of which is in turn composed of three other phases, making a total of nine steps to complete the total cycle.

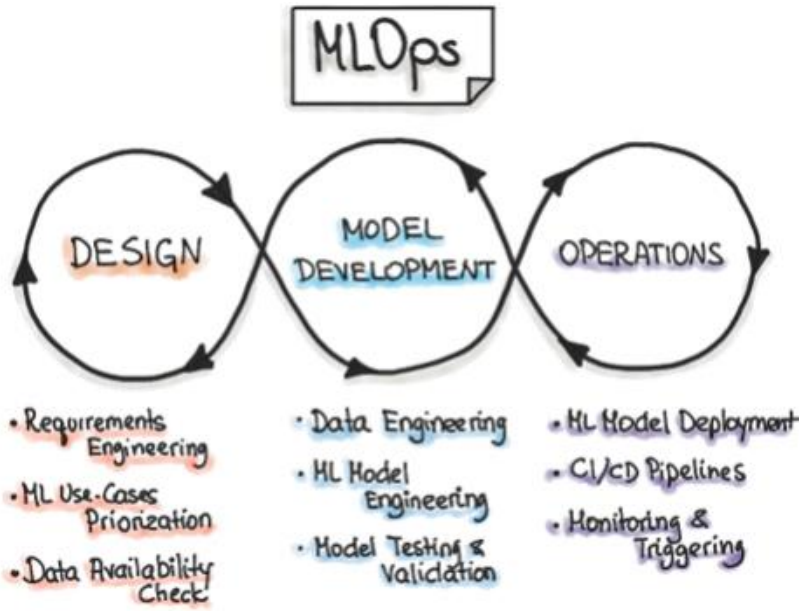

Figure 1 - View of the nine stages of the MLOps lifecycle (retrieved from https://ml-ops.org/content/mlops-principles)

The lifecycle of MLOps encompasses the end-to-end management and optimisation of ML models and workflows, integrating DevOps and data science practices. The lifecycle always begins with the problem definition and data collection, where the business goals and relevant data sources are identified. This is followed by data preprocessing, including cleaning, transformation, and feature engineering, to ensure the data is ready for modelling. The next phase involves model development, exploring and evaluating various algorithms and techniques. Once a suitable model is selected, it undergoes training and validation using historical data. After model training, the focus shifts to deployment and monitoring. The model is deployed to production environments where it interacts with real-time data, and monitoring tools are put in place to track its performance, detect anomalies, and ensure reliability.

3.2. Tools and Solutions for the MLOps Lifecycle

In Table 1 below which is a compilation we have made concerning existing frameworks and tools used to complete each of the nine steps we have mentioned. Afterwards, we explain in more detail each of these nine phases.

Table 1 - Summary of Tools and Solutions for the MLOps Lifecycle

Requirements Engineering		
Purpose	To identify, analyse, document, and manage the needs and expectations of stakeholders for a software development project	
Examples		
JIRA	This is a widely used project management tool that also includes features for requirements management	https://www.atlassian.com/software/jira
Confluence	This is a wiki-based collaboration tool that can be used for requirements management	https://www.atlassian.com/software/confluence
Visual Paradigm	This is a modelling tool that includes features for requirements engineering, such as the ability to create use cases, user stories, and requirements diagrams	https://www.visual-paradigm.com/
Diagrams.net	Good general purpose Technical Diagram tool to help standardise, between partners, the format of architectural, process diagrams and layouts	https://app.diagrams.net/
ML Use cases prioritisation		
Purpose	The process of identifying and prioritising potential use cases for applying machine learning algorithms in a particular domain or industry	
Examples		
DataRobot	Automated machine learning platform that includes features for use case prioritisation. It uses a combination of artificial intelligence and human expertise to identify and prioritise potential use cases	https://www.datarobot.com/
Microsoft Azure ML Studio	It includes tools for identifying and prioritising use cases	https://azure.microsoft.com/en-us/products/machine-learning/
Rapid Miner	It includes tools for evaluating and selecting the most promising use cases for implementation	https://rapidminer.com/

Data Availability Check		
Purpose	To ensure that the necessary data is available and accessible for use in a machine learning project	
Examples		
SageMaker Data Wrangler	A tool that provides a user-friendly interface for data cleaning and preparation	https://aws.amazon.com/sagemaker/data-wrangler
Trifacta	Data preparation platform that provides features for data profiling, cleaning, and enrichment	https://www.trifacta.com
Azure Data Factory	Data integration platform that provides features for data preparation, transformation, and validation	https://azure.microsoft.com/en-us/products/data-factory
Industreweb	Large library of Protocol shopfloor edge connectivity, can be used to verify if a suitable protocol connector is available, and if not then add a new type to support the pilot requirements	https://www.industreweb.co.uk/
Data Engineering		
Purpose	To design, build, and maintain the infrastructure required to support the collection, storage, processing, and analysis	
Examples		
Apache Spark	Open-source data processing engine that provides features for batch processing, streaming, and machine learning	https://spark.apache.org/
Apache Kafka	Distributed streaming platform that provides features for data processing and messaging	https://kafka.apache.org/
Apache Hadoop	Open-source framework that provides features for distributed storage and processing of large data sets	https://hadoop.apache.org/
Industreweb	Edge Router with connectivity of vast majority of shopfloor protocols and near real-time processing of data for the provision of data to ML components	https://www.industreweb.co.uk/
ML Model Engineering		
Purpose	To design and develop machine learning models that can perform complex tasks with high accuracy and reliability	

Examples		
Google Cloud AutoML	Tools for automated ML (AutoML), including AutoML Vision, AutoML Video Intelligence, AutoML Natural Language, and AutoML Translation	https://cloud.google.com/automl
H2o.ai	Open-source platform that provides several tools for AutoML	https://h2o.ai/
TPOT	Open-source AutoML tool that automates the process of building and optimising machine learning pipelines	http://automl.info/
Model Testing and Validation		
Purpose	To evaluate the performance and accuracy of a machine learning model and to ensure that it can generalise well to new data	
Examples		
Scikit-learn	Python library for machine learning that provides a wide range of tools for model selection, evaluation, and validation	https://scikit-learn.org/
Keras	Deep learning API that provides a range of tools for model training, evaluation, and validation	https://keras.io/
PyTorch	Open-source machine learning framework that provides a range of tools for model training, evaluation, and validation	https://pytorch.org/
ML Model Deployment		
Purpose	To integrate a trained machine learning model into a production environment so that it can be used to make predictions or decisions based on new data.	
Examples		
Kubeflow	Open-source platform for deploying and managing machine learning workflows on Kubernetes	https://www.kubeflow.org
Docker	Platform for building, packaging, and deploying applications in containers	https://www.docker.com/
Microsoft Azure Machine Learning	Cloud-based platform for building, training, and deploying machine learning models	https://azure.microsoft.com/en-us/products/machine-learning/
CI/CD pipelines		

Purpose	The purpose of continuous integration and continuous deployment (CI/CD) pipelines is to automate the machine learning model development and deployment process	
Examples		
Jenkins	Open-source automation server that provides a wide range of features	https://www.jenkins.io/
GitLab CI/CD	Platform for continuous integration and continuous deployment that provides features for building, testing, and deploying machine learning models	https://docs.gitlab.com/ee/ci/
CircleCI	Cloud-based continuous integration and continuous deployment platform	https://circleci.com
Monitoring and Triggering		
Purpose	To continuously monitor the performance of machine learning models deployed in production environments and trigger actions when necessary	
Examples		
Prometheus	Open-source monitoring system that provides features for monitoring and alerting on various aspects of the machine learning pipeline	https://prometheus.io/
Grafana	Open-source platform for data visualisation and monitoring that can be used to create dashboards and alerts for monitoring machine learning models	https://grafana.com/
Kibana	Open-source platform for data visualisation and analysis that can be used to monitor and analyse the performance of machine learning models	https://www.elastic.co/es/kibana/
Industreweb	Industreweb Collect Engine allows for detection of events and the triggering of mitigating actions. Industreweb Display dashboards allow for screens to be created to reflect ML status	https://www.industreweb.co.uk/

As can be seen, some tools allow the completion of one or several phases. Moreover, this whole ecosystem is constantly evolving. This means that some tools might fall into disuse, and others emerge again, so it is essential to refer to the concepts that tend to be more immutable over time. Therefore, it is necessary to remark on the need for a strategic and adaptable approach to tool selection and utilisation.

The idea behind recognising the temporary nature of these tools is to facilitate stakeholders to ensure sustained maintenance leading to continuous innovation within their organisations.

3.3. AI REDGIO 5.0 SME's AI/ML Needs Identification

As a first approach for figuring out the requirements of our platforms, we have used a questionnaire for collecting information about industrial pilot experiments related to AIREDGIO 5.0. To do that, we have compiled structured questions within sections to gather information about each industrial pilot, its objectives, and its use of AI technologies. This questionnaire allows us to gather information from the different pilot partners in a standardised format.

The different sections we are interested in are the following:

- **Experiment Profile:** Gather basic information about the pilot application, its title, responsible partner, description of the use case, and the date of the bilateral call.
- **Brief description of the use case:** Describe the problem, objectives, and current solutions (if any) related to the industrial pilot experiment.
- **AI Aspects of the Experiment:** Detailing how AI is expected to be used in the experiment, including data collection, analysis, and actionable predictions.
- **Related AI Perspective:** Identifying the AI technologies or perspectives relevant to the experiment.
- **Applicable AI Technologies:** List specific AI technologies that may apply to the industrial pilot.
- **Collaborative Intelligence:** Describing how humans and AI will work together in the experiment.
- **Suitable Edge AI Paradigm:** Identifying the edge AI paradigm that should be supported for the experiment.
- **ML-Ops Lifecycle:** Providing information about the status of the different steps in the machine learning operations lifecycle for the experiment.
- **Technical Info:** Collecting technical details related to data aspects, sources, frequency, processing type, format, volume, and delivery mechanism.
- **Assigned technology provider in AIREDGIO 5.0:** Identifying any technology or data science partners involved in the experiment.
- **Personnel with relevant expertise:** Determining if the pilot partner has personnel with relevant experience for the experiment.

This questionnaire served as a structured way to gather comprehensive information about each industrial pilot participating in the AIREDGIO 5.0 project. It further allowed for a standardised and organised approach for collecting data about the experiments and their AI-related aspects and helping the requirements analysis of the experiment with regards to their AI needs and the incorporation of Human in the Loop (HITL) aspects in the use case.

4. Collaborative Intelligence Platform for Industry 5.0

Technological advancements have consistently reshaped a wide range of industries, leading to the evolution of industrial processes. Industry 4.0 was a significant milestone by leveraging cyber-physical systems and data-driven automation to facilitate manufacturing processes [1]. However, as the world moves towards a more connected future, the emergence of Industry 5.0 and Edge AI presents a paradigm change that needs further exploration. We aim to introduce the principles, benefits, and challenges associated with Collaborative Intelligence within the framework of Industry 5.0 in the present technological landscape.

Industry 5.0 represents the convergence of automation technologies and human-centric approaches to create a symbiotic relationship between humans and machines [2]. Unlike its predecessor, which primarily focuses on optimising efficiency through automation, Industry 5.0 emphasises human workers' unique cognitive abilities [3]. Our work aims to further investigate the potential of human-machine collaboration using human expertise and problem-solving skills in combination with the precision and speed of machines.

Collaborative Intelligence approaches have already become valuable strategies for improving the performance of various natural language processing (NLP) tasks, such as machine translation and text summarisation. Similarly, CI approaches have been used in computer vision to augment the capabilities of object detection and image classification tasks. In healthcare, CI methodologies have facilitated the development of critical applications such as medical diagnosis and drug discovery. CI approaches hold significant promise for elevating the performance of ML models. The idea of combining the strengths of human expertise and computational algorithms, should facilitate enhancing the potential to achieve results that would otherwise remain unachievable through human or machine-driven processes in isolation [4].

In ML methodologies, active learning is a prominent approach where human involvement is pivotal in discerning the most instructive data samples, thus enabling the model to refine its learning process. A complementary technique, known as iterative learning, operates through cyclic iterations of the model's execution, during which human evaluators offer feedback. This feedback catalyses improving the model's performance over subsequent iterations.

Additionally, the so-called human-in-the-loop approach reveals another aspect of interactive machine learning, wherein human operators engage in a dynamic interaction with a ML model specifically designed to govern a physical system. In this interactive arrangement, humans' real-time feedback assumes a substantial role in the model's decision-making processes, influencing and optimising the system's performance.

In the literature, there are several classical strategies for CI mentioned. Since these tasks are so common in many work teams, it is often overlooked that they are a basic form of CI.

- **Data labelling:** People often label data for ML models. This is especially important for tasks that are difficult or time-consuming for machines, such as identifying objects in images or extracting text from documents.
- **Model training:** People can help to train ML models. This can be done by providing feedback to the model during training or by manually adjusting the model's parameters.
- **Model deployment:** People can also deploy ML models in the real world. This can involve monitoring the model's performance and adjusting as needed.

As we have said, these are basic forms of CI, and later we will see some more advanced forms. For example, CI can address complex challenges in natural language processing, computer vision, healthcare, and more. Integrating CI methodologies like active learning, iterative learning, and human-in-the-loop control

allows ML models to continuously improve and adapt, achieving unprecedented levels of performance and efficiency [5].

4.1. Human – AI CI Landscape Analysis

We have seen that CI is a human-machine partnership that combines the strengths of both humans and AI to achieve better results than either could on their own. CI systems are designed to be co-creative, meaning humans, and AI work together to solve problems and make decisions. On the other hand, Industry 5.0 is a new industrial revolution focused on human-centric manufacturing. In this new era, AI can augment human capabilities rather than replace them. CI is essential for Industry 5.0 because it allows humans and AI to work together effectively.

We have already seen the classic CI techniques in which a human operator (often a data scientist) was in collaboration with the machine, but now let us look at the situations whereby a human operator might need to be involved:

- **Machine learning with human feedback:** This is a type of CI where humans provide feedback to AI systems to help them learn and improve. For example, humans might be asked to label data or correct errors in the AI system's output. The clearest example of this category is the HITL solution.
- **Co-creation:** This type of collaborative intelligence is where humans and AI systems work together to create new products or services. For example, humans might provide AI systems with their ideas and feedback, while AI systems provide humans with access to new data and insights. The clearest example of this category would be the emerging Large Language Models (LLMs)
- **Augmented intelligence:** This type of CI uses AI systems to augment human capabilities. For example, AI systems might provide real-time assistance to doctors, help pilots fly planes or any other related activity.

However, although the literature is gradually offering more innovative solutions to these categories, there are also still some important challenges to overcome:

- **Cost and Scalability:** Integrating humans can increase operational expenses, and scalability may become challenging.
- **Training and Expertise:** Ensuring human contributors have the necessary training and expertise to provide accurate input to AI systems.
- **Workflow Integration:** Establishing seamless workflows between humans and AI systems, minimising delays and bottlenecks.
- **Balancing Human-AI Interaction:** Striking the right balance between human involvement and AI automation, optimising system performance.

As if that is not enough, these new strategies also have additional and important requirements. For example:

- **The need for trust:** Humans and AI must be able to trust each other to work together effectively.
- **The need for data:** CI systems require a lot of data to learn and improve. In some cases, we are talking about very gigantic amounts of data needed to train the systems, which is not always possible.
- **The need for ethical guidelines:** CI systems must be developed and used ethically and in accordance with the laws.

In addition to these challenges, another important aspect of CI is the need for continuous communication and understanding between humans and AI systems. Effective collaboration relies on clear communication and mutual comprehension of goals and expectations. Human operators must be able to interpret and understand the outputs and decisions made by AI systems, while AI systems should be able to accurately comprehend human intentions and context [6].

Although a few recognise the difficulties of implementing these approaches, it is important to remark that there are several advantages associated to a successful strategy realised when these are used.

- **Improved Accuracy**: Human expertise can help correct AI mistakes and enhancing overall system performance.
- **Ethical Decision-making**: Humans can provide context, empathy, and ethical judgment in complex situations where AI alone may struggle.
- **Adaptability and Flexibility**: HITL allows AI systems to learn and adapt quickly, incorporating new information and addressing unforeseen scenarios.
- **Data Quality Assurance**: Humans can validate and curate datasets, ensuring high-quality inputs for AI training and avoiding biases.

Therefore, one of the most vital challenges in this context is facilitating effective communication and understanding between humans and AI systems. Clear and transparent interaction becomes essential, as both parties must comprehend each other's intents and expectations. This bidirectional dialogue should facilitate a collaborative innovation towards progress. In the literature, a non-exhaustive set of best practices is recognised, including but not limited to:

- Clearly defining the human roles and responsibilities within the interaction process between people and machines
- Establishing proper feedback mechanisms for human-AI interaction, enabling continuous learning and improvement.
- Investing in training and education for human contributors to ensure proficiency and knowledge.
- Regularly evaluating and updating the collaborative process to adapt to evolving needs and technology advancements.

In the following subsections, we will look at concrete examples of software solutions that facilitate collaboration between operators and machines in the manufacturing context.

4.1.1. Orchestra

Orchestra[1] is a software platform that enables the integration of different IT systems. It is designed to be a low-code platform, which means that it can be used by non-technical users to create integrations. Orchestra can be used to enable real-time data exchange between different systems. This is useful for applications requiring up-to-date data, such as manufacturing control and supply chain management systems.

Orchestra is intended to serve as a bridge between siloed systems, enabling a collaborative environment where data flows without restrictions. This orchestration is intended to enhance productivity and reduces manual intervention, minimising errors and simplifying processes.

[1] https://orchestral.ai/

Using a HITL system like Orchestra will improve project the management processes, resulting in increased accuracy and consistency, less time coordinating projects, and more expert time in the areas they are uniquely positioned to work on.[2]

In addition, Orchestra's flexibility extends beyond data integration. It also offers various customisable automation capabilities, allowing human operators to streamline complex workflows and decision-making processes. In this context, this solution can intelligently route tasks, prioritise workloads, and adapt to changing conditions in real time. This adaptability enhances efficiency and empowers organisations to stay agile and responsive in fast-paced landscapes. In summary, whether orchestrating data or tasks, this solution offers a new level of efficiency and collaboration.

4.1.2. Teaming.AI

Teaming.AI[3] is a cutting-edge platform designed to foster collaborative interactions between human stakeholders and artificial intelligence systems within the context of Industry 4.0 [7]. The primary objective of Teaming.AI is to address the limitations of current Industry 4.0 practices, particularly the lack of flexibility, by establishing a human-centred AI collaboration model. This approach seeks to ensure that humans remain in control and autonomous while leveraging the capabilities of AI to achieve more efficient and effective industrial processes.

The platform encompasses a comprehensive suite of software components that facilitate seamless interactions between the Teaming.AI system and all human-AI team members. At its core, the platform is tailored to cater to specific industrial facilities, referred to as the target system, which hosts various production processes, such as manufacturing and quality control of products [8].

Teaming engineers play a pivotal role in implementing and maintaining the Teaming.AI platform. Their responsibilities include connecting the target system to the platform and ensuring its smooth operation. To accomplish this, they engage in various tasks, including teaming modelling, knowledge management, deployment of machine learning algorithms, and ongoing maintenance of the overall teaming platform [9]. The teaming engineers come from diverse backgrounds, such as process designers, data engineers, AI experts, or knowledge engineers, bringing together a multidisciplinary approach to address the complexities of the industrial domain.

Operators, on the other hand, form an integral part of the human-AI team, actively participating in the use case-specific activities enabled by the Teaming.AI platform. Their involvement entails executing specific teaming processes, providing crucial feedback to enhance the Teaming.AI platform's performance, and interacting directly with the target system. Referred to as HITL, operators' involvement ensures that human expertise and decision-making remain central to the industrial processes, even as AI is integrated into the workflow.

The human stakeholders are classified into two groups based on their roles within the Teaming.AI ecosystem. First, the teaming engineers, also known as Human-on-the-loop (HOTL), shoulder responsibilities related to the design, management, and continuous improvement of the Teaming.AI platform. Their activities are primarily focused on the technical aspects of the system, ensuring its adaptability and responsiveness.

The human operators actively participate in the daily operations of the target system. Their tasks involve:

[2] https://www.b12.io/orchestra/

[3] https://www.teamingai-project.eu/

- interacting with the Teaming.AI platform to execute predefined teaming processes,
- offering feedback to refine the AI algorithms, and
- engaging with the physical processes within the industrial facility.

The division of roles between these two groups is flexible and depends on the use case, with each group having specific and complementary responsibilities.

Figure 2 shows an example of the Teaming.AI platform (in Spanish) where the human operator has the chance to inform the system that the quality control of a newly generated piece was not satisfactory. To do so, the human operator has the option of pressing Yes (the green button) or No (the red button). The samples that the operator does not approve are reused to recalibrate the model so that you have a feedback mechanism that is very beneficial in most cases.

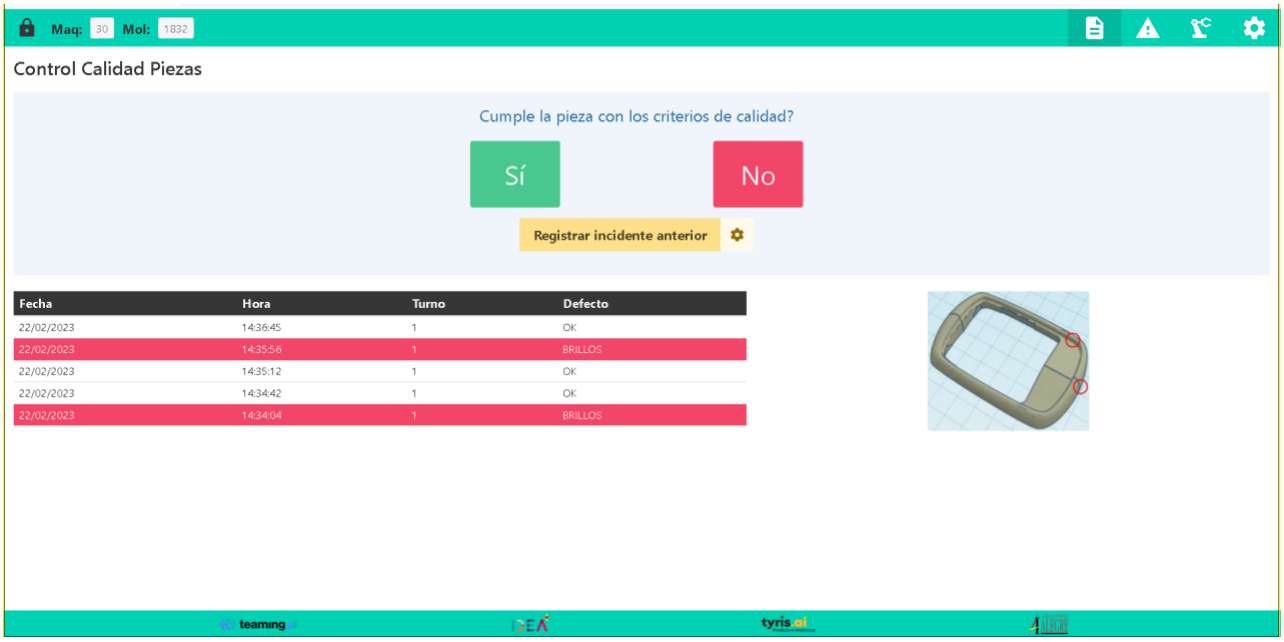

Figure 2 - Teaming.AI Interface for Human Operators - Confirmation of the Successful Completion of Parts

In conclusion, Teaming.AI is an innovative platform that empowers human-AI collaboration in Industry 4.0 settings. By integrating human expertise with AI capabilities, the platform aims to enhance industrial processes, promote flexibility, and maintain human autonomy in the future of industrial scenarios.

The platform's holistic approach encompasses all aspects of software development, target system interaction, and user engagement. It is a comprehensive and adaptive solution for various use cases within the Industry 4.0 landscape.

4.2. AI REDGIO 5.0 Collaborative Intelligence Platform for Industry 5.0

The AI REDGIO 5.0 Collaborative Intelligence Platform is a solution at the forefront of Industry 5.0 [10]. The idea is to combine various technological advancements to redefine industrial landscapes. The platform facilitates Human-AI collaboration by integrating cutting-edge AI capabilities. In this way, the platform is intended to illustrate the potential of connected devices, sensors, and machines through real-time data fusion and analysis, driving optimal decision-making and resource allocation.

4.2.1. Requirements

The most important high-level requirements for AI REDGIO 5.0 Collaborative Intelligence Platform for Industry 5.0 are:

- **Requirement 1. Integration of Advanced Technologies:** The platform integrates cutting-edge technologies, including AI-driven analytics, and Internet of Things (IoT)-enabled devices to create a synergistic ecosystem.
- **Requirement 2. Human-Machine Interaction:** The platform integrates the capability of interaction between human operators and machines.
- **Requirement 3. Real-time Data Analytics:** The platform conducts real-time data analytics on the vast volumes of data collected from IoT devices.
- **Requirement 4. Continuous Learning and Knowledge Management:** The platform stores a central knowledge-sharing and learning management repository.
- **Requirement 5. Collaborative Innovation:** The platform facilitates a culture of collaborative innovation, where workers can virtually collaborate to propose process improvements and product enhancements.

4.2.2. Usage Walkthrough

Our platform adopts a practical problem-solving approach, data analysis, and process optimisation. This subsection is about a comprehensive usage walkthrough, explaining each step's significance.

- **Step 1: Onboarding and User Authentication:** This step should ensure secure access to the system, safeguarding sensitive data and mitigating potential security breaches.
- **Step 2: Integrating IoT and Smart Devices:** This step should involve the integration of IoT and smart devices using a methodology for the interconnection of heterogeneous devices.
- **Step 3: AI-driven Data Analytics and Insights:** The system can start working to detect patterns, anomalies, and trends within the data through ML algorithms.
- **Step 4: Human-Machine Interaction and Augmentation:** This step should involve the coordination of actions between human operators and machines, emphasising augmentation rather than substitution.
- **Step 5: Real-Time Process Optimisation:** This step should be about to automatically adjusting operational parameters in response to changing conditions.
- **Step 6: Knowledge Base and Learning Management:** This step should be about organising a repository to capture organisational know-how, accumulated insights, and best practices.
- **Step 7: Collaborative Innovation and Continuous Improvement:** This step should facilitate cross-functional collaboration and iterative enhancement through data analytics, human-machine collaboration, and organisational knowledge.

Figure 3 shows an example in the form of a diagram flow of the usage walkthrough. Please note that not all steps are mandatory. For example, Step 6 for the creation of a knowledge base with the operation history, although desirable, does not have to be used in all scenarios.

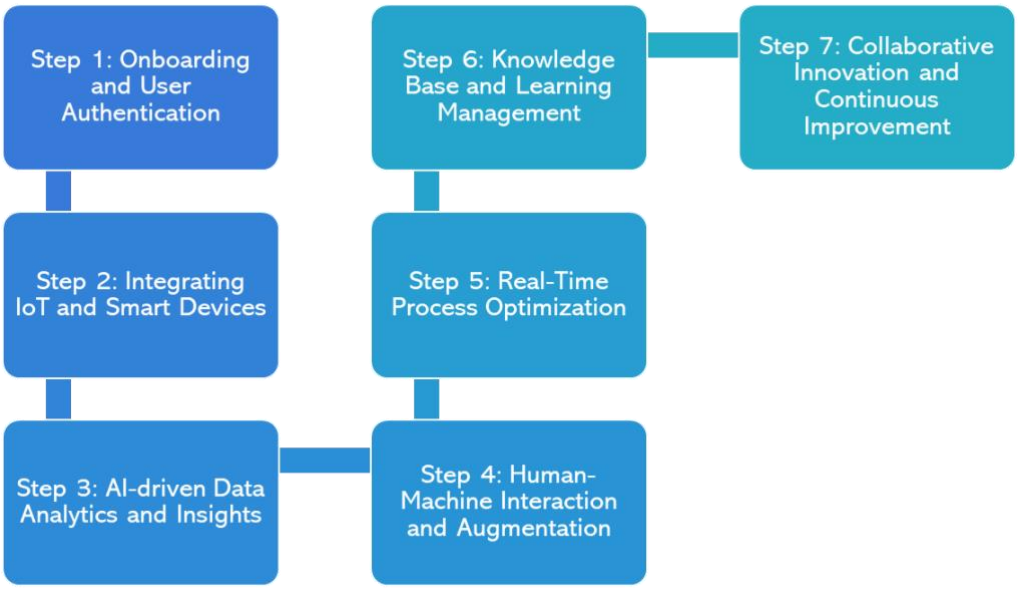

Figure 3 - Collaborative Intelligence Platform for Industry 5.0 - Usage Walkthrough

4.2.3. CI Platform Upcoming Work

To date, we have been working on requirements gathering and solution analysis. We have also partially addressed essential design issues. The next steps are to finalise the design and integrate the various components to form a fully integrated solution. During these first phases, we have prioritised the creation of a horizontal solution, which can be used in all (or at least most) of the pilot projects we have analysed. The next steps, therefore, has to do with the assembly, implementation of the solution and testing with the pilots.

The components to be assembled in this phase are:

- The CI component that allows the human operator to accept or reject a manufactured product. This component has an interface that can be configured around various metrics including, but not limited to, accuracy, interpretability, speed, etc.
- The pipeline creation component that allows us to easily create, configure, and deploy workflows in various work environments. This component acts as the backbone of a robust and efficient workflow management system, enabling organisations to optimise their processes and achieve higher productivity.
- The interfacing component that connects the operator interface with hardware and housing pre-trained models ready for deployment in manufacturing environments. This component acts as the bridge between the digital and physical realms of our experiments.

After this integration, we will start with the testing phase, before having everything ready to move on to the realisation of the experiments.

5. Open Hardware Platform for Embedded AI and AI-at-the-Edge

The Open Hardware Platform for Embedded Artificial Intelligence and AI-at-the-Edge represents a significant advancement in technology and artificial intelligence (AI). This platform is based on open hardware, which allows users and developers to modify and enhance hardware according to their specific needs. Open Hardware contrasts with proprietary hardware, which is closed and cannot be changed by users.

Embedded AI as a term refers to placing AI execution directly into edge devices, such as IoT devices, rather than relying on a connection to a central server or the cloud for AI processing, allowing edge devices to perform AI tasks, such as image processing, speech recognition and decision-making, autonomously and in real-time.

The Open Hardware Platform for Embedded AI and AI-at-the-Edge have a wide range of applications; it can be used in autonomous drones for image processing and real-time decision-making, in personal assistance devices for voice recognition and real-time interaction, or industrial sensors for monitoring and independent decision making.

5.1. Embedded AI and AI-at-the-Edge Landscape Analysis

It is important to consider the background and state-of-the-art technologies in different areas to develop the technological asset in Artificial Intelligence on open-source hardware.

In the following, each of the above aspects will be addressed:

5.1.1. Architectures and Standards

TensorFlow[4], PyTorch[5], and Keras[6] are among the most widely used architectures for developing machine learning for ESP32[7]. These frameworks offer a set of tools and APIs that facilitate the implementation of Machine Learning and Deep Learning algorithms on the ESP32. These architectures provide great flexibility and power to develop AI models on the board, allowing neural networks to be trained and executed efficiently.

In addition, norms and standards such as ONNX[8] (Open Neural Network Exchange) have been established to facilitate the exchange of AI models between different frameworks and platforms. AI models developed in compatible frameworks can be converted to ONNX format and run on the ESP32. This interoperability is especially valuable, allowing previously trained models to be leveraged and shared across different environments.

5.1.2. Data Models and Ontologies

Data models refer to the structure and organisation of the data used to train and feed Machine Learning algorithms. In the case of ESP32, various data formats, such as tensors and matrices, can be used to represent features and training labels. This data can come from sources such as sensors connected to the ESP32, databases, or even real-time data collection provided by other devices or sensors.

In the context of Machine Learning on ESP32, ontologies can be used to describe and organise the relationships between data and entities of the problem at hand. For example, an ontology can be created in natural language to build language-independent representations that can serve as a meeting point between

[4] https://www.tensorflow.org/
[5] https://pytorch.org/
[6] https://keras.io/
[7] https://en.wikipedia.org/wiki/ESP32
[8] https://onnx.ai/

two or more natural languages. In this sense, an ontology is considered the repository of concepts that establish connections between the symbols of a language and their referents in the world, the possible states and actions that can be performed. These ontologies help to improve the understanding of the problem and can be used to infer conclusions or make decisions based on the represented knowledge.

5.1.3. Data Management Solutions

Data Quality

Data quality is a crucial aspect in developing Machine Learning models, as the results obtained are largely dependent on the data quality used for training and evaluating the model.

Essential concepts about data quality:

- Accuracy and integrity of data: Data must be accurate and truthful, meaning that it must be free of errors, inconsistencies, and biases. Errors in the data can introduce noise and negatively affect model performance.
- Complete data contain all the variables and information needed for the problem. Missing data or missing values can hinder the proper training of the model and affect its performance.
- Data should be consistent in terms of format, unit of measurement, and structure. Inconsistencies can make it challenging to interpret the data correctly and affect the quality of the model.
- Data should be relevant to the problem being addressed. Including irrelevant or noisy data can negatively affect the model's ability to learn meaningful patterns and produce accurate results.
- In classification problems, it is essential to ensure that classes are balanced regarding the number of examples available. An imbalance in the classes can lead to a bias in the model towards the majority class and affect its ability to generalise correctly.
- In the case of labelled datasets, the quality of the labelling is critical. Incorrect or ambiguous labelling can affect the quality of the model and generate erroneous results.
- Before training a model, it is common to perform data cleaning and pre-processing tasks to remove noise, eliminate outliers, normalise scales, etc. These techniques help to improve the quality of the data and to prepare them properly for the model.

In short, data quality is essential for the success of Machine Learning models. Data must be accurate, complete, consistent, relevant, and balanced. In addition, correct labelling and proper pre-processing are key aspects to ensure the quality of the data used in training and evaluating models.

5.1.4. Data Processing Architecture (Edge vs. Cloud)

Edge Data Processing

Edge data processing refers to performing data processing and analysis on devices or systems at the network's edge, close to the source of data generation. This means data is processed locally on the device before being sent to a remote data centre or the cloud.

Advantages of Edge Processing

- Low latency: By processing data at the edge, response time is reduced, which is crucial for applications that require fast responses.

- Privacy and security: Data is kept locally and not sent to the cloud, which can be beneficial for ensuring the confidentiality and protection of sensitive data.
- Reduced network load: By performing processing at the edge, the amount of data sent over the network is reduced, reducing congestion and bandwidth costs.

Limitations of Edge Processing

- Limited resources: devices at the edge, such as sensors or IoT devices, may have limitations in memory, processing power, and storage.
- Limited scalability: Due to resource constraints, the ability to scale edge processing may be limited.

Cloud Data Processing

Cloud data processing involves sending data to remote servers located in larger, more powerful data centres for processing and analysis. The results or inferences can be returned to the device at the edge for action or decision-making.

Advantages of Cloud Processing

- Scalable resources: Cloud services offer large processing and storage capacity, allowing scaling according to the system's needs.
- Flexibility and agility: Cloud processing enables sophisticated analytics and Machine Learning techniques to be applied to large volumes of data.
- Collaboration and centralisation: By processing data in the cloud, multiple devices and systems can access and collaborate in real time, enabling distributed applications.

Limitations of Cloud Processing

- Latency: Due to the need to send data over the network to the cloud, there can be increased latency compared to edge processing.
- Connectivity dependency: Cloud architecture requires a reliable Internet connection for processing and data transmission.

5.1.5. Data Analytics (approach and tools)

There are different approaches and tools to perform data analytics in AI. Some of the standard techniques include:

- Data pre-processing: Data may require cleaning, normalisation, and transformation to ensure quality and consistency before analysis.
- Exploratory data analysis: Visualisation and summary statistics techniques are used to understand the structure and distribution of data, identify outliers and detect relationships between variables.
- Machine Learning: Machine learning algorithms are used to train data-driven models and perform tasks such as classification, regression, clustering, and anomaly detection.

- Deep Learning: A branch of machine learning that uses deep neural networks to learn complex features and patterns in data, enabling more sophisticated tasks such as image recognition, natural language processing, and autonomous driving.
- Natural language processing (NLP): Used to analyse and understand human language, enabling information extraction and unstructured text processing.
- Data mining: Data mining techniques are applied to discover hidden patterns and practical insights in large datasets, which can provide valuable information for decision-making.

It is important to note that the success of data analytics in AI depends on the quality of the data used, the appropriate selection of techniques and algorithms, and the correct interpretation of the results. In addition, the advancement in data processing technologies and the availability of large amounts of data have driven the growth and application of data analytics in artificial intelligence.

5.1.6. Data Sovereignty Solutions

Data sovereignty requires ensuring the privacy and protection of sensitive data while complying with security regulations and standards. This implies using architectures and encryption techniques that safeguard data confidentiality.

The Artificial Intelligence at the Edge approach helps to reduce the risk of data leakage by keeping data on devices located in private networks with limited external access. This approach provides greater control and security over data by processing it locally before it is sent over the network.

5.1.7. Interoperability Layer and APIs

ESP32, as a development platform, has a wide range of libraries that facilitate communication with other devices and services. These libraries allow connections to be established using different technologies, providing flexibility and versatility to projects. Some of the available options include:

- REST APIs[9] over WIFI: Libraries such as WiFiClient[10] and ESPAsyncWebServer[11] can implement HTTP-based application programming interfaces (APIs) over WIFI connections. This allows communication and data exchange between the ESP32 and other devices via HTTP requests and responses.
- Queuing services such as MQTT[12]: MQTT (Message Queuing Telemetry Transport) is a lightweight and efficient messaging protocol ideal for communication between IoT devices. Libraries such as PubSubClient allow the ESP32 to act as both a publisher and a subscriber, sending and receiving messages through an MQTT broker.
- LoRa[13]: The ESP32 also supports LoRa (Long Range) technology, which enables long-range, low-power communication. Libraries such as LoRaLib[14] provide functions to configure and use the LoRa module built into the ESP32, enabling wireless communication over significant distances.

[9] https://www.redhat.com/en/topics/api/what-is-a-rest-api
[10] https://www.arduino.cc/reference/en/libraries/wifi/wificlient/
[11] https://github.com/me-no-dev/ESPAsyncWebServer
[12] https://mqtt.org/
[13] https://docs.arduino.cc/learn/communication/lorawan-101
[14] https://github.com/jgromes/LoRaLib

- GSM/GPRS/LTE[15]: For projects requiring cellular connectivity, libraries such as TinyGSM[16] allow the ESP32 to communicate with mobile networks via GSM, GPRS, or LTE modules. These libraries facilitate the configuration of cellular connectivity and the exchange of data with remote services.
- Bluetooth: The ESP32 supports Bluetooth, enabling wireless communication with nearby devices. Libraries such as ESP32 BLE Arduino[17] provide functions to configure and use Bluetooth connectivity on the ESP32, allowing interaction with other Bluetooth devices such as sensors, actuators, or smartphones.

5.2. AI REDGIO 5.0 Open Hardware Platform for Embedded AI and AI-at-the-Edge

In the context of AI REDGIO 5.0, existing background knowledge and experience brought by HOPU/Libelium, in the area of edge AI for smart cities solutions, will be further utilised to fit the needs of the manufacturing sector and the AI REDGIO 5.0 experiments.

In the context of the initial work performed in T5.2 so far, an example AI model for urban air quality has been tested on the Open Hardware Platform. The role of this example is twofold: a. it acts as an initial proof-of-concept of the platform's capabilities, and b. it is also available to any user that want to test the Open Hardware Platform with an initial example before deploying their own manufacturing-related models.

The following sections present the aforementioned work, including the specifications for the deployment of an AI model on the Open Hardware Platform, the steps towards the deployment of a model on the Open Hardware platform, and the AI Deployment Specifications on the Open Hardware.

The complete process for deploying AI models on open-source development boards such as ESP32 follows the following steps:

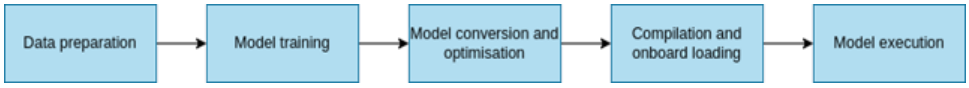

Figure 4 - General Architecture for Open Hardware AI Deployment

- **Data preparation**: This step involves collecting and preparing the data needed to train and evaluate the AI model. This may include collecting relevant datasets, labelling them, and splitting them into training, validation, and test sets.
- **Model training**: Using a suitable development environment, such as TensorFlow, PyTorch, or Keras, the AI model is trained using the prepared dataset. During this process, parameters are adjusted, and iterations are performed to achieve an optimal model.
- **Model conversion and optimisation**: Once the model has been trained, it must be converted to a format compatible with the ESP32 development board. This may involve altering the model to a format such as TensorFlow Lite, suitable for running on resource-constrained devices. In addition, optimisation techniques can be applied to reduce the model's size and improve its efficiency.
- **Compilation and onboard loading**: The next step is to compile the converted model for the specific ESP32 architecture. This may require using tools like the Arduino integrated development

[15] https://www.4gltemall.com/blog/what-is-gsm-edge-gprs-umts-3g-hsdpa-hsupa-lte/
[16] https://github.com/vshymanskyy/TinyGSM
[17] https://www.arduino.cc/reference/en/libraries/esp32-ble-arduino/

environment (IDE) or custom development platforms. Once compiled, the model is loaded onto the ESP32 development board.

- **Model implementation**: Once the model has been successfully loaded onto the ESP32 board, it can be used for real-time inference. This involves providing new input data to the model and obtaining predictions or results based on the trained model.

It is important to note that the process of deploying AI models on open-source development boards such as ESP32 can vary depending on the tools and technologies used, as well as the specific requirements of the project. However, these steps provide an overview of the process generally followed to deploy AI models on these development boards.

It should also be noted that there are some limitations, such as TensorFlow instructions that cannot be used when deploying models on Micro Controller Units (MCUs[18]), as well as incompatibilities between different versions if we use the method described in this document; this is described further below.

5.2.1. Proof of concept for validating the approach

The main proof of concept to validate the progress of this task has been performed through the successful integration of a rudimentary artificial intelligence (AI) model on the Open Hardware Platform, with the ESP32 development board as the basis.

This example is a proof of concept and is an example of developing AI models for manufacturing by using an artificial intelligence model to make air quality predictions on the data that a sensor would send via MQTT, as shown in Figure 5.

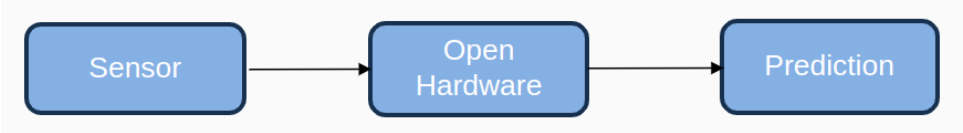

Figure 5 - High Level Component Architecture

To achieve this, the first step is to generate an artificial intelligence model that is capable of making the necessary predictions. In the context of the current example, the model used is an AI model designed and developed as a tailor-made tool to predict air quality parameters. Going deeper, its capabilities extend to predicting the concentration of ozone (O3) that could be observed in the next hour. This prediction is based on a comprehensive analysis of reported values of particulate matter such as PM10 and PM2.5, gases such as nitrogen dioxide (NO2), sulphur dioxide (SO2), ozone (O3) and carbon monoxide (CO).

Once the phases of data preparation, model training and model generation and optimisation have been passed, the last two steps, generate the Open Hardware code, compilation and finally the deployment on hardware, are still to be tested on real Open Hardware.

For this purpose, the configurations made for the realisation of this proof of concept are described below.

5.2.2. Open Hardware Deployment Steps

Following are the steps required by the Open Hardware Platform users in order to deploy their AI models on the platform.

[18] https://en.wikipedia.org/wiki/Microcontroller

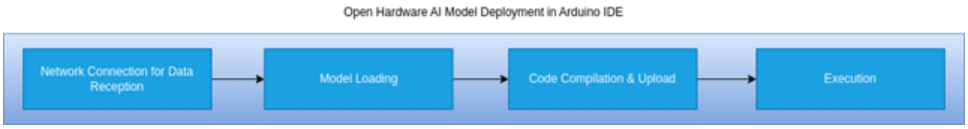

Figure 6 - AI Deployment in the Open Hardware

Network Connection and Data Reception

The ESP32 board, known for its versatility, has been configured to establish a connection to a Wi-Fi network thanks to its capabilities as a formidable Internet of Things (IoT) device.

In our quest for seamless data reception, we have incorporated the PubSubClient[19] library. This library is critical to the board's ability to receive data via the MQTT protocol, which is renowned for its lightness and efficiency, especially in low-power devices. However, our team is also exploring the potential of a REST API, which could pave the way for receiving data via a more conventional HTTP network protocol and could facilitate its implementation within other frameworks on the Edge.

Model Loading and Prediction

The next phase is to take advantage of the TensorFlowLite ESP32 library[20], a feature available in the Arduino Integrated Development Environment[21] (IDE). This library plays a crucial role in the initialisation of the essential variables to load the AI model on the board and guarantee its perfect execution, obtaining the same output that would be obtained by running the model on a more powerful computer.

Compilation and Code Uploading to the Development Board

The culmination of any development process is the deployment of the code onto the target device. After meticulous configurations and ensuring that every aspect of the code aligns with the example requirements, the next step is the compilation. This is where the raw code is transformed into a format that the development board can understand and execute. The Arduino IDE, known for its user-friendly interface, simplifies this process. Instead of navigating through complex command lines or scripts, the IDE provides a straightforward "Upload" button. A single click on this button initiates the process, transferring the compiled code onto the development board.

Execution

With the code securely in place, the board transitions from a dormant state to an active one, eager to perform its tasks. In this context, the board is designed to receive data via MQTT, a lightweight messaging protocol optimised for low-bandwidth, high-latency networks.

To monitor the board's activities and ensure its functioning as expected, the Arduino IDE provides another invaluable tool: the Serial Monitor, shown in Figure 7. This tool provides a window into the board's operations, displaying real-time data and messages time. Opening the Serial Monitor allows one to observe the data the board processes, ensuring its accuracy and timeliness.

[19] https://www.arduino.cc/reference/en/libraries/pubsubclient/

[20] https://github.com/tanakamasayuki/Arduino_TensorFlowLite_ESP32

[21] https://en.wikipedia.org/wiki/Integrated_development_environment

Figure 7 - Arduino IDE Serial Monitor

To activate the AI model embedded in the board, one does not need to go through complex procedures to activate the AI model embedded in the board. A simple MQTT message, sent to the topic to which the board is subscribed, is sufficient. Figure 8 shows a message used during this proof of concept.

```
→ ~ docker run -it --rm efrecon/mqtt-client pub -h broker.emqx.io -t "topicName/aq" -m \
'{"PM10":156.0, "PM2.5":137.0, "NO2":29.0,"SO2":17.0,"O3":5.0,"CO":1.1}'
```

Figure 8 - MQTT Message

Once the board receives this message, it triggers the AI model, which processes the data and performs its predictive functions. This seamless integration of hardware and software, combined with user-friendly tools, ensures that even complex tasks like AI predictions become accessible and straightforward. The Prediction of the AI Model can be seen in Figure 9.

Figure 9 - AI Model Prediction Output

Importance of Version Consistency

A fundamental aspect that cannot be overlooked is the version consistency of TensorFlow. The version used during the training and validation phase of the AI model must reflect the version used during its deployment on the ESP32 board. If the Arduino IDE's TensorFlow Lite library for ESP32 is chosen, it is essential to use TensorFlow version 2.1.1. Any deviation from this could lead to discrepancies, manifesting as errors during model loading or subsequent execution.

5.2.3. Open Hardware System and Software Specifications

This section provided the hardware and software specifications of the ESP32, which will be used in AI REDGIO 5.0 as the reference open hardware platform that will support the implementation of the pilot experiments.

Open Hardware Specifications

ESP32-Wrover-B contains two low-power Xtensa® 32-bit LX6 microprocessors. The internal memory includes:

• 448 KB of ROM for booting and core functions.

• 520 KB of on-chip SRAM for data and instructions.

• 8 KB of SRAM in RTC, which is called RTC FAST Memory and can be used for data storage; it is accessed by the main CPU during RTC Boot from the Deep-sleep mode.

• 8 KB of SRAM in RTC, which is called RTC SLOW Memory and can be accessed by the co-processor during the Deep-sleep mode.

• 1 Kbit of eFuse: 256 bits are used for the system (MAC address and chip configuration) and the remaining768 bits are reserved for customer applications, including flash-encryption and chip-ID.

Software Requirements

Deploying an artificial intelligence model on an ESP32 requires a combination of specific tools and libraries, both for model development and hardware implementation.

The main software requirements are detailed below:

- Development Environment:
  - Arduino IDE or PlatformIO: Both are popular development environments for programming the ESP32. Arduino IDE is more beginner friendly, while PlatformIO offers more advanced features.
- AI Libraries and Tools:
  - TensorFlow Lite for Microcontrollers: This is a version of TensorFlow designed for low-power devices such as the ESP32. It allows you to convert TensorFlow models into formats that can be run on microcontrollers.
  - ESP32 TensorFlow Lite Arduino Library: A library that facilitates the integration of TensorFlow Lite models into Arduino projects for the ESP32.
- ESP32 Drivers and Libraries:
  - ESP32 Board Manager and ESP32 Libraries: These are needed to program and communicate with the ESP32 from the Arduino IDE.
  - PubSubClient (optional): If you plan to integrate the ESP32 with MQTT for IoT communications.
- Modelling and Training Tools:

- TensorFlow: The primary tool for designing, training and converting AI models for use on microcontrollers.
- Python: TensorFlow and many other AI-related tools are based on Python, so it is essential to have a proper installation of Python and pip (Python package manager).

- Conversion Tools:
  - TensorFlow Lite Converter: once the AI model has been trained with TensorFlow, it needs to be converted to a format that is compatible with TensorFlow Lite for Microcontrollers.
  - Additional Dependencies (depending on the project):
    - Libraries for specific sensors or actuators if they are involved in the project (e.g. temperature sensors, cameras, motors, etc.).
    - Communication libraries if specific connectivity is required (e.g. Wi-Fi, Bluetooth, LoRa, etc.).
- Debugging and monitoring tools:
  - Serial Monitor: Included in the Arduino IDE and PlatformIO, it allows to monitor the program output in real time, which is essential for debugging.

It is noted, that in case other hardware than ESP32 is selected, then in terms of hardware specification that should be able to satisfy the requirements imposed by the software tools presented in the next list.

5.3. Open Hardware Upcoming Work

To summarise, our journey has seen the successful deployment of a fundamental AI model on an ESP32 board. This example holds promise for future examples to be developed during the execution of the project to serve the needs of the AI REDGIO 5.0 experiments. However, while we rejoice in the success of this application, it is essential to remain aware of the inherent limitations of the ESP32 board, especially regarding its processing prowess and memory capacity. These limitations highlight the need for further refinement and optimisation of the AI model to ensure that its operation remains smooth and efficient.

For Task 5.2 the next main objective at this stage is to explore and understand how the acquired knowledge can be implemented on edge frameworks. The selection of the most suitable framework will be crucial and will be determined following a thorough analysis of the available options and their respective compatibilities and capabilities.

In collaboration with the other AI REDGIO 5.0 partners, the design and development of various tasks and experiments will be undertaken. This collaboration aims to combine the practical experience and theoretical knowledge of both entities to create robust and efficient solutions. The expected outcome of this collaboration is the development and deployment of a demonstration illustrating the application of an innovative Artificial Intelligence (AI) model in the field of manufacturing, using Open Hardware.

Although the specific use cases to be addressed by the AI models is yet to be determined, these will be for sure use cases that are in the heart of the manufacturing domain. The specific tasks related to the design, development and deployment of the models will be further defined as the project progresses (with the

models developed under WP6 and the deployment done in WP5), ensuring that each step is aligned with the overall project objectives and contributes significantly to its success.

To this end, an analysis of the needs of one exemplary use case (with the collaboration of Polimi) will be carried out to deploy it using one of the production lines available in its facilities, as well as the edge framework that best suits this situation. The synergy between the project team and the Polimi will be vital for the effective development of the AI model and its subsequent deployment in the selected environment to act as an exemplary case that will then be used by the other experiments to design and develop their own AI models.

The resulting of this exemplary use case demo will serve as a concrete example of the potential of AI integration in manufacturing through the use of open hardware, providing a solid foundation for future research and development in the area.

6. Conclusions and Next Steps

The AI REDGIO 5.0 Collaborative Intelligence Platform (T5.1) and the Open Hardware Platform (T5.2) constitute two of the core Edge AI tools developed in the context of WP5. The former enables end-users integrate into their AI solutions the aspect of Human-in-the-Loop, while the latter facilitates users leverage on the computational and storage capabilities of their edge devices, with the use of open-source solutions to deploy and execute locally AI and ML pipelines and models, for enhanced security and speed.

This first deliverable (D5.1) sets the foundations for the design and development of these solutions, from the scope of landscaping on the adjacent technologies and paradigms, such as collaborative intelligence, edge and embedded AI, AI-ML Ops, as well as from the users' scope in the context of the AI REDGIO 5.0 project. Bilateral interviews have been performed with the AI REDGIO 5.0 SMEs for an initial extraction of cloud-edge AI and CI experiments' needs, resulting in the initial set of requirements and specifications for the CI Platform and the Open Hardware Platform. These specifications served as the basis on the one hand for the design of the mock-ups of the Collaborative Intelligence Platform and on the other hand for the experimentation and early proof-of-concept with the deployment of real-life AI models on the Open Hardware Platform. The early results of the first version of the platforms designed in the context of T5.1 and T5.2 are publicly available through the early deployment of the Edge AI Reference Library, under the URL http://wiki.ai-redgio50.s5labs.eu/ . This online repository will serve as the point-of-reference for the exploration of AI resources developed within or relevant to the AI REDGIO5.0 project and will be continuously updated to include the latest advancements and versions of the WP5 tools.

The next steps for the platforms presented in this deliverable D5.1 entail the following advancements:

- Collaborative Intelligence Platform:
  - finalisation of design
  - integration of subcomponents, and namely: the CI component facilitating rejection or acceptance of a manufacturing product by the human operator; workflows pipeline creation component, interfacing component
  - testing and use in the context of AI REDGIO 5.0 experiments
- Open Hardware Platform:
  - selection and experimentation of the interplay between the Open Hardware Platform and Edge Frameworks.
  - design and implementation of demonstration on the POLIMI facilities, illustrating the application of an innovative Artificial Intelligence (AI) model in the field of manufacturing, using Open Hardware

Afterwards the tools will undergo a technical verification process in the context of the integration activities of T5.6. The feedback and results of the verification process, along with the progression of scenarios and user requirements in WP2 and WP6 will result in the identification and implementation of the final version of the tools, that will be delivered in the context of D5.4 [M27].

7. References

[1] Rožanec, J. M., Novalija, I., Zajec, P., Kenda, K., Tavakoli Ghinani, H., Suh, S., ... & Soldatos, J. (2023). Human-centric artificial intelligence architecture for industry 5.0 applications. International Journal of Production Research, 61(20), 6847-6872.

[2] Lindner, F., & Reiner, G. (2023, May). Industry 5.0 and Operations Management—the Importance of Human Factors. In NOMS 2023-2023 IEEE/IFIP Network Operations and Management Symposium (pp. 1-4). IEEE.

[3] Ordieres-Meré, J., Gutierrez, M., & Villalba-Díez, J. (2023). Toward the industry 5.0 paradigm: Increasing value creation through the robust integration of humans and machines. Computers in Industry, 150, 103947

[4] Loizaga, E., Eyam, A. T., Bastida, L., & Lastra, J. L. M. (2023). A Comprehensive study of human factors, sensory principles and commercial solutions for future human-centered working operations in Industry 5.0. IEEE Access.

[5] Dimou, A. (2022, September). On a generalized framework for time-aware knowledge graphs. In Towards a Knowledge-Aware AI: SEMANTiCS 2022—Proceedings of the 18th International Conference on Semantic Systems, 13-15 September 2022, Vienna, Austria (Vol. 55, p. 69). IOS Press.

[6] Biffl, S., Hoffmann, D., Kiesling, E., Meixner, K., Lüder, A., & Winkler, D. (2023, June). Validating Production Test Scenarios with Cyber-Physical System Design Models. In 2023 IEEE 25th Conference on Business Informatics (CBI) (pp. 1-10). IEEE.

[7] Haindl, P., Hoch, T., Dominguez, J., Aperribai, J., Ure, N. K., & Tunçel, M. (2022, September). Quality Characteristics of a Software Platform for Human-AI Teaming in Smart Manufacturing. In International Conference on the Quality of Information and Communications Technology (pp. 3-17). Cham: Springer International Publishing.

[8] Hoch, T., Heinzl, B., Czech, G., Khan, M., Waibel, P., Bachhofner, S., ... & Moser, B. (2022). Teaming. AI: enabling human-AI teaming intelligence in manufacturing. Proceedings http://ceur-ws. org ISSN, 1613, 0073.

[9] Bachhofner, S., Kiesling, E., Revoredo, K., Waibel, P., & Polleres, A. (2022, July). Automated process knowledge graph construction from BPMN models. In International Conference on Database and Expert Systems Applications (pp. 32-47). Cham: Springer International Publishing.

[10] Ivanov, D. (2023). The Industry 5.0 framework: Viability-based integration of the resilience, sustainability, and human-centricity perspectives. International Journal of Production Research, 61(5), 1683-1695.

ANNEX – Pilot Interviews Questionnaire

AI REDGIO 5.0 Industrial Pilots - Edge AI Needs Collection

Pilot Name:

Partner Name:

Use case Name:

Date of Bilateral Call:

Experiment Profile

Please note answers in bullet-like format

1. Brief description of use case (driving problem(s), experiment objective(s), current solution (if applicable)): *A brief description of the industrial pilot experiment*.
2. Envisioned use of AI in the AIREDGIO5.0 Experiment: *A brief description of the expected solution from the industrial pilot's perspective, e.g. "an intelligent learning approach for product quality prediction in production line X. The AIREDGIO5.0 solution will allow to early detect defects and failures, leveraging deep learning technologies."*

AI Aspects of the Experiment

1. Expectations from AI in the AIREDGIO 5.0 Experiment: *A brief description of how the AI solution is expected to work in the industrial pilot's settings, e.g. real-time (or batch daily) data will be collected from production line X, they will be analysed in the AIREDGIO 5.0 cloud infrastructures (or edge infrastructures of the industrial pilot) and actionable predictions (in natural language) will be provided every 30'/hour/day to the factory workers.*
2. Related AI Perspective: *High-level reference to the AI technology(-ies) that will be used in the experiment. Multiple can apply. If pilot partner doesn't know, answer to be extracted by WP5 partners. Indicative list of perspectives:*
    a. Computer Vision
    b. Machine Learning
    c. Expert Systems
    d. Speech
    e. Natural Language Processing (NLP) & Generation (NLG)
    f. Robotics & Automation
    g. Recommendation Engines
    h. Other (please define): Optimisation
3. Applicable AI Technologies: *Depending on the AI Perspective that has been selected, different AI technologies will be put into use. In case any such technologies have been already prioritised by the industrial pilot, they should be highlighted. Some indicative AI technologies that may apply to the industrial pilots are the following (list non-exhaustive, options not mutually exclusive). If pilot partner doesn't already know, answer to be extracted by WP5 partners. Indicative list of technologies:*

a. Natural Language Processing (NLP) & Generation (NLG): Digital Assistants Chatbots, Content/Semantic Recognition, Media and Language Knowledge Extraction, Content Generation, Natural Language Synthesis
b. Computer Vision: Machine Vision, Image/Face/Video Recognition, Image/Video Processing
c. Machine Learning: Deep Learning, Supervised Learning, Unsupervised Learning, Semi-supervised Learning, Reinforcement Learning, Self-Supervised Learning, Evolutionary Learning
d. Expert Systems: Rule-based Systems, Decision Support Systems
e. Speech: Speech-to-Text Technologies, Text-To-Speech Technologies
f. Robotics & Automation: Self-Learning Robots, Artificially Intelligent Cobots, Connected & Automated Vehicles
g. Recommendation Engines: Collaborative Filtering Techniques, Content-based Filtering Techniques, Hybrid Filtering Techniques

4. Collaborative Intelligence in the AIREDGIO 5.0 Experiment: A brief description of how the aspect of collaborative intelligence (i.e. humans and AI actively enhancing each other's complementary strengths) will be addressed/demonstrated in the experiment.
5. Suitable Edge AI Paradigm: the edge AI paradigm that should be supported for the purposes of the experiment. If pilot partner doesn't already know, answer to be extracted by WP5 partners. Indicative answers:
    a. Embedded Machine learning (EML)
    b. TinyML
    c. Federated Machine Learning
    d. Cloud-edge AI
    e. Other (please define)
6. ML-Ops Lifecycle and AIREDGIO 5.0 experiment: steps of the ML-Ops Lifecycle in AI REDGIO 5.0. Indicate current status (pending, ongoing, completed, blocked) for experiment. Status to be updated periodically.

    Definitions for each step:
    a. Requirements Engineering: To identify, analyse, document, and manage the needs and expectations of stakeholders for a software development project
    b. ML Use cases prioritisation: The process of identifying and prioritising potential use cases for applying machine learning algorithms in a particular domain or industry
    c. Data Availability Check: To ensure that the necessary data is available and accessible for use in a machine learning project
    d. Data Engineering: To design, build, and maintain the infrastructure required to support the collection, storage, processing, and analysis
        → ML Model Engineering: To design and develop machine learning models that can perform complex tasks with high accuracy and reliability
        → Model Validation: To evaluate the performance and accuracy of a machine learning model and to ensure that it can generalise well to new data
    e. ML Model Deployment: To integrate a trained machine learning model into a production environment so that it can be used to make predictions or decisions based on new data.

f. CI/CD pipelines: The purpose of continuous integration and continuous deployment (CI/CD) pipelines is to automate the machine learning model development and deployment process
g. Monitoring and Triggering: To continuously monitor the performance of machine learning models deployed in production environments and trigger actions when necessary.

Date of Update	Requirements Engineering	ML Use cases prioritisation	Data Availability Check	Data Engineering	ML Model Engineering	Model Testing and Validation	ML Model Deployment	CI/CD pipelines	Monitoring and Triggering

Technical Information

1. Data Aspects: The data aspects relevant to the experiment

Date of Update	Data Sources (sensors, historical data, other)	Data Frequency (how often will new data be available)	Data Processing Type (real-time, batch, other)	Data Format (structured text, unstructured text, image, video, other)	Anticipated Volume	Data Delivery Mechanism (through API, File, other)	Current Data Availability Status (available, pending, other)

2. Assigned technology provider in AIREDGIO 5.0: is there any tech/data science partner that will support the implementation of the experiment?
3. Any data scientist or other contact point with relevant expertise in the industry pilot partner team: Does the pilot partner have any personnel with relevant experience that will be involved in the experimentation?