



Article

Fostering Agricultural Transformation through AI: An Open-Source AI Architecture Exploiting the MLOps Paradigm

Antonio Carlos Cob-Parro ^{1,2,*} , Yerhard Lalangui ¹ and Raquel Lazcano ¹

¹ Atos Research and Innovation, Albarracín 25, 28037 Madrid, Spain; yergard.lalangui@atos.net (Y.L.); raquel.lazcano@atos.net (R.L.)

² University of Alcalá, Department of Electronics, Ctra. Madrid-Barcelona, km. 33600, 28805 Alcalá de Henares, Spain

* Correspondence: antonio.cob@atos.net or antonio.cob@edu.uah.es

Abstract: As the global population is expected to reach 10 billion by 2050, the agricultural sector faces the challenge of achieving an increase of 60% in food production without using much more land. This paper explores the potential of Artificial Intelligence (AI) to bridge this “land gap” and mitigate the environmental implications of agricultural land use. Typically, the problem with using AI in such agricultural sectors is the need for more specific infrastructure to enable developers to design AI and ML engineers to deploy these AIs. It is, therefore, essential to develop dedicated infrastructures to apply AI models that optimize resource extraction in the agricultural sector. This article presents an infrastructure for the execution and development of AI-based models using open-source technology, and this infrastructure has been optimized and tuned for agricultural environments. By embracing the MLOps culture, the automation of AI model development processes is promoted, ensuring efficient workflows, fostering collaboration among multidisciplinary teams, and promoting the rapid deployment of AI-driven solutions adaptable to changing field conditions. The proposed architecture integrates state-of-the-art tools to cover the entire AI model lifecycle, enabling efficient workflows for data scientists and ML engineers. Considering the nature of the agricultural field, it also supports diverse IoT protocols, ensuring communication between sensors and AI models and running multiple AI models simultaneously, optimizing hardware resource utilization. Surveys specifically designed and conducted for this paper with professionals related to AI show promising results. These findings demonstrate that the proposed architecture helps close the gap between data scientists and ML engineers, easing the collaboration between them and simplifying their work through the whole AI model lifecycle.

Keywords: Agriculture; Artificial Intelligence; Machine Learning; Deep Learning; Internet of Things; DevOps; MLOps



Citation: Cob-Parro, A.C.; Lalangui, Y.; Lazcano, R. Fostering Agricultural Transformation through AI: An Open-Source AI Architecture Exploiting the MLOps Paradigm. *Agronomy* **2024**, *14*, 259. <https://doi.org/10.3390/agronomy14020259>

Academic Editors: Sara Álvarez and Sergio Vélez

Received: 13 December 2023

Revised: 15 January 2024

Accepted: 22 January 2024

Published: 25 January 2024



Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

According to a United Nations (UN) report (<https://www.un.org/development/desa/en/news/population/world-population-prospects-2019.html>, accessed on 12 December 2023), the global population is expected to grow to almost 10 billion people by 2050. Considering this estimation, the World Resources Institute (WRI) calculates that, to keep up with this demand, food production will have to be increased by at least 60% in the same period [1]. However, to cope with that need, the agricultural land that would be needed exceeds by far the land available as of today. This is a concept referred to by the WRI as *land gap*, and it has towering climate implications: using that much land for agricultural purposes would destroy vital ecosystems, which in turn would contribute even more to climate change, with food production being nowadays already responsible for almost 25% of global greenhouse gas emissions [1].

In recent years, many lines of research have started to explore the possibility of solving the land gap problem via the application of cutting-edge technologies, making agriculture

itself evolve toward a new era, which has already been called, amongst others, *smart farming*, *precision agriculture*, or *Agriculture 4.0* [2]. This field of study has gained a lot of momentum in the last few years since it is considered one of the key contributors toward the UN's Sustainable Development Goals (<https://sdgs.un.org/goals>, accessed on 12 December 2023) introduced in 2015 in the 2030 Agenda for Sustainable Development (<https://sdgs.un.org/2030agenda>, accessed on 12 December 2023).

Undoubtedly, Artificial Intelligence (AI) is a pivotal technology in the realm of smart agriculture, as noted in the works [3,4]. Its integration has revolutionized this field, opening up new avenues to enhance both the quantity and quality of crop yields, as well as to automate processes. This paves the way for intelligent, autonomous systems that can learn and make informed decisions. For instance, the study in [5] highlights how AI can reduce chemical use by up to 90%, showcasing its efficacy in optimizing agricultural production processes, leading to more efficient farming, increased productivity, and reduced environmental impact.

In addition, Shankar et al. (2020) [6] present insights on how AI-driven strategies can refine crop protection, bolstering sustainable agriculture. This research demonstrates the significant improvements AI can bring to crop management strategies. Further exploring AI's role in environmentally conscious agriculture, the authors in [7] review precision chemical weed management strategies and propose a new CNN-based modular spot sprayer. This innovation is a testament to how AI can be applied to develop more precise and efficient weed control solutions, reducing the overall chemical footprint in farming.

Furthermore, the work carried out in [8] delves into how AI can alleviate environmental challenges posed by agriculture. It provides a comprehensive look at how AI can be implemented for more efficient crop production and monitoring while minimizing ecological footprints. Complementing these findings, Visentin et al. (2023) [9] investigate a mixed-autonomous robotic platform for precise weed removal in both intra-row and inter-row settings. This development underscores the role of AI in precision agriculture, illustrating how robotic systems can be specialized for tasks like exact weed control, which helps in reducing chemical herbicide reliance and supports sustainable agriculture.

Additionally, the deployment of collaborative smart robots, as detailed in the work [10], represents a significant advancement. In this case, a group of robots leverages AI to optimize harvesting routes, thereby boosting crop collection volumes. This not only exemplifies the increasing autonomy in agricultural systems but also their efficiency and ecological responsibility.

The integration of AI in agriculture, as mentioned, seeks not only to enhance productivity, but also to ensure the welfare and efficiency of both the machinery and the workforce, emphasizing the potential of AI in aiding human workers rather than substituting them. Implementing AI in conjunction with UGVs, for example, to assist workers in optimizing fruit harvesting, or to accurately distribute phytosanitary products, highlights how well humans and machines can work together. Such synergies not only optimize agricultural processes but also ensure the protection of the environment and the sustainability of resources. This approach seeks a balance, where technology serves human efforts rather than replacing them, ensuring that the insights and expertise of human intervention remain an integral part of the process.

However, this inclusion comes with its own share of challenges to overcome before starting to consider its extensive adoption within the agricultural domain.

The enormous development and extension of AI in recent years is undeniable [11–13]. It has spread so much that it has become a revolution, being applied in almost every sector imaginable: embedded finances [14], business value [15], transport management [16], medicine [17], industry 4.0 [18], and, of course, agriculture [19–21].

Precisely, agriculture is one of the domains where AI is gaining much importance. The automation of processes through Machine Learning (ML) or Deep Learning (DL) based models is expected to allow for the substitution of humans by machines (UGVs, UAVs), to perform repetitive and costly tasks. This, in turn, is expected to increase the

performance and efficiency of the task at hand. However, the application of AI in production or operational environments in general, and in the agriculture domain in particular, still faces many challenges. One of the main open issues in AI right now is not so much the creation of AI models themselves, but their deployment in production environments, their maintenance throughout their entire life cycles, and the management of the huge datasets that are usually involved. Rapid changes in models and data need continuous updates in production systems. Asset management, including model versions and data, should be autonomous and optimal as much as possible. The true challenge of AI integration lies in adapting to increasingly rapid changes, optimizing that integration for real-world scenarios, and maintaining an organized workflow to implement these measurements. Right now, there is not a single solution for this, but rather a plethora of tools that AI practitioners such as data scientists or ML engineers need to master before even getting to use them, let alone considering their usage in industrial environments. In the current environment, it is up to the data scientist/ML engineer to study the different tools and assess which ones are most suitable to build the whole workflow/model life cycle. This makes the learning curve a steep one and hinders the acceptance of AI in environments where it creates complete disruption, generating distrust in a technology that is seen as a black box, such as agriculture.

To address these issues, a new paradigm known as Machine Learning Operations (MLOps) has emerged. Its objectives are twofold: (1) automating the process of building ML models and deploying them to production; and (2) maintaining and monitoring these models throughout their whole life cycle to detect potential issues which could compromise the AI model's performance and automate a response [22–24], increasing the efficiency and scalability, as well as reducing the potential risks. By embracing MLOps culture, developers unlock the advantages of optimized workflows, automated AI model deployment, and effective collaboration, leading to increased productivity and robustness, faster development cycles, and better performance of their AI models.

Therefore, the objective of this paper is to provide a solution for some of the aforementioned open challenges. We propose an open-source AI architecture based on the MLOps paradigm to reduce the complexity of developing and deploying AI models in agricultural contexts. The proposed architecture seeks to improve upon the state-of-the-art MLOps methods by implementing a functional and tested architecture that is used by several AI stakeholders. This solution aims to (1) minimize the learning curve associated with managing AI models without a centralized MLOps platform and (2) promote the acceptance of AI in agriculture by presenting an integrating approach to develop and deploy AI models, store datasets, and even gather data from different sources of information. It supports state-of-the-art IoT communication protocols such as Message Queuing Telemetry Transport (MQTT) [25] and Hypertext Transfer Protocol (HTTP) [26].

Hence, the main contributions of this paper are the following:

- An AI architecture using open-source technologies for creating and producing AI models is presented, covering the whole life cycle of the AI model, from its creation to its deployment and monitoring.
- The architecture builds a workflow made of state-of-the-art tools that enable data scientists and ML engineers to work more efficiently and rapidly, solving many problems in their day-to-day lives.
- The architecture supports the access through different types of IoT protocols, such as HTTP and MQTT, enabling ease of access and communication with diverse devices.
- The system is able to run different AI models at the same time, making optimal use of the hardware resources available in the cluster where the platform has been deployed.

The rest of the document is organized as follows. First, in Section 2.1, the related work is analyzed and presented. Then, the proposed architecture with its different components, which is the main contribution of this paper, is described in Section 2.2. Insights gathered from key stakeholders who have already been exposed to the platform are summarized in Section 3. Finally, the main conclusions are extracted in Section 5.

2. Materials and Methods

This section covers two essential elements of our research. Section 2.1 “Related Work” presents an in-depth analysis of existing studies, establishing the background and highlighting potential contributions of our research. Section 2.2 “Proposed Architecture” outlines our distinctive strategy, describing the technologies and methods applied to demonstrate both the novelty and practicality of our proposed solution.

2.1. Related Work

The integration of AI and other advanced technologies into agriculture represents a significant paradigm shift in one of the most essential industries. This transformation is driven by the convergence of various technological innovations, including ML, computer vision, Internet of Things (IoT), and big data analytics. Together, these technologies are reshaping agricultural practices, enabling more efficient resource management, enhancing productivity, and contributing to sustainability. However, the adoption and implementation of these technologies present unique challenges and opportunities. This section provides a review of the current related work in AI and technology integration in agriculture, exploring key developments, challenges, and future prospects.

2.1.1. Artificial Intelligence in Agriculture

Agriculture, one of the oldest and most vital industries, has been transformed by the integration of AI. Computer vision systems are now capable of detecting pests and diseases with remarkable accuracy [27]. Predictive algorithms have been developed to forecast weather patterns, enabling farmers to make informed decisions [28,29]. However, the implementation of AI in agriculture is not without challenges. Data variability, lack of standardized datasets, and resistance in traditional farming communities have slowed progress [30–34]. Collaborative efforts between AI experts and agronomists are essential to bridge this gap [35].

In addition to developing algorithms and Artificial Intelligences to increase field productivity, improving the efficiency of agricultural practices is crucial. This perspective is vital in addressing climate change and the environmental challenges observed in the last decade. Therefore, the digitization and application of AI in agricultural settings introduce new challenges, including data management, the utilization of crop-specific technologies, and the need for new infrastructure to process these data [36].

Another critical area where Artificial Intelligence is extensively used to mitigate environmental implications is in water management, which is one of the most critical factors in crop cultivation. Intelligent water management, administered precisely and efficiently, ensures that crops are adequately hydrated to yield the maximum amount of food while conserving water usage [37]. However, this raises the same dilemma as previously mentioned. It necessitates a network of sensors measuring various environmental parameters, communicating these parameters to an infrastructure for data collection and processing, ultimately determining the opening of water valves. To address this need, this article proposes a platform enabling interaction with the environment through the use of AI models.

2.1.2. Open-Source Architectures and MLOps

The open-source movement has democratized access to cutting-edge tools, with platforms such as TensorFlow [38] and PyTorch [39] leading the way. This democratization has led to the emergence of MLOps, a practice that seeks to standardize and automate the AI model lifecycle [40,41]. In agriculture, MLOps enables rapid deployment of solutions, adapting to changing field conditions [42]. It also fosters collaboration among multidisciplinary teams, ensuring that models are developed and maintained efficiently [43,44].

The field of MLOps, now integral to AI development, has found a significant application in agriculture. MLOps are dedicated to streamlining and automating the complete AI model lifecycle, encompassing development, deployment, and ongoing maintenance.

In agriculture, this translates to the ability to swiftly deploy AI-based solutions capable of adapting to ever-changing field conditions.

Efficient model deployment holds immense value in agriculture, where real-time decision making is paramount for maximizing productivity and resource efficiency. For instance, computer vision algorithms for early disease detection can trigger rapid, precise responses, potentially averting significant crop losses [45].

Some studies have concentrated on the MLOps paradigm from a theoretical point of view [46], while others have presented various tools which could be applied in an MLOps platform [40,47]. Additionally, several studies have implemented portions of the overall solution [48,49]. However, there is a noticeable gap in the MLOps literature, providing a solution that integrates all these components, offering a first approach for a functional and tested MLOps architecture.

2.1.3. Smart Agriculture and Agriculture 4.0

Agriculture 4.0 is the next frontier, representing the convergence of digital and physical technologies in the agricultural sector [50,51]. IoT sensors are now used to monitor soil moisture and autonomously activate irrigation systems [52,53]. Robotics has found applications in tasks such as harvesting and weeding [54]. Big data analytics and AI are being used to optimize resource allocation and increase productivity [55–57]. However, interoperability between different devices and platforms remains a significant challenge, requiring further research and standardization [58,59].

In recent years, the concept of Agriculture 4.0, often referred to as “Smart Agriculture”, has gained substantial attention in the research and industry communities. This transformative approach harnesses the power of digital technologies to address various challenges in agriculture. One of the key aspects of Agriculture 4.0 is the utilization of IoT devices and sensors for real-time monitoring and control of agricultural processes. These devices collect data on soil conditions, weather patterns, crop health, and equipment status. With the help of AI algorithms, these data are processed to make informed decisions regarding irrigation, pest control, and crop management [60,61].

Furthermore, the integration of robotics in agriculture has marked a significant advancement. Autonomous drones and robotic systems are employed for tasks such as precision planting, harvesting, and weed control. These technologies not only enhance the efficiency of operations but also reduce the need for manual labor, addressing labor shortages in agriculture [62,63]. Moreover, the use of robotics contributes to minimizing the environmental impact by enabling targeted application of resources [64].

The role of big data analytics cannot be overstated in the context of Agriculture 4.0. Large volumes of data are generated from various sources, including sensors, satellites, and machinery. Advanced analytics techniques, such as Machine Learning and data mining, are applied to extract valuable insights from these data. These insights enable farmers and agricultural professionals to make data-driven decisions that optimize crop yield, resource utilization, and sustainability practices [55–57].

2.1.4. User Experience in Agricultural Systems

The integration of technology into agriculture must be user-centric. Systems must be adapted to work in remote areas with limited connectivity [65,66]. Interfaces have to be designed to be used without requiring deep technical knowledge [67]. Training and support are crucial components to ensure that technological solutions are adopted and used effectively in the field [68]. Moreover, cultural and socioeconomic factors must be considered to create solutions that are truly aligned with the farmers needs [41,45].

Effective training and ongoing support are indispensable components of successful technology adoption in agriculture. Farmers and agricultural workers need proper training to harness the full potential of technology-enabled solutions [68]. This includes not only understanding how to operate the technology but also interpreting the data it provides.

Support mechanisms, whether through local agricultural extension services or remote tech support, should be readily available to address issues and provide guidance as needed.

Agricultural technology solutions should be designed with a deep understanding of the cultural and socioeconomic factors that influence farming practices. Different regions and communities have unique needs and preferences, which should be taken into account during the development of agricultural systems [45]. Solutions that align with local customs and preferences are more likely to be embraced by the farming community.

2.1.5. Recent Developments and Future Directions

Recent developments in AI and technology have opened new paths for innovation in agriculture. Advanced weather prediction models are providing more accurate forecasts, allowing for better planning and resource management [29]. New methodologies are being developed to address data variability and standardization challenges [31]. Community engagement strategies are being explored to overcome resistance in traditional farming regions [34]. Collaborative AI development is fostering a more inclusive and efficient approach to technological innovation [44].

Edge computing, an emerging paradigm in agriculture, holds significant promise for enhancing real-time data processing and decision making at the farm level. By deploying edge devices equipped with AI capabilities, such as field-based sensors and edge servers, farmers can analyze data directly at its source. This approach reduces latency and allows for quicker responses to changing environmental conditions [61]. Edge computing is particularly beneficial for applications such as precision irrigation, where timely data insights can optimize water usage and crop health. As edge computing technologies continue to mature, their integration into agricultural systems is expected to grow, further improving farm efficiency and sustainability.

In light of increasing climate variability and extreme weather events, resilience in agriculture has become a paramount concern. AI-driven solutions are aiding farmers in adapting to these challenges. Machine Learning models can analyze historical weather data and predict potential climate-related risks, enabling farmers to implement mitigation strategies and protect their crops [60]. Furthermore, AI-powered drones equipped with thermal imaging cameras can identify stress in crops caused by heatwaves or water shortages, allowing for targeted interventions [63]. Building climate-resilient agricultural systems through AI technologies is vital for ensuring food security in the face of a changing climate.

The global agricultural community is increasingly recognizing the importance of collaboration and data sharing in addressing common challenges. Initiatives such as the Agricultural Model Intercomparison and Improvement Project (AgMIP) are bringing together researchers, farmers, and policymakers to share data and develop models that enhance agricultural sustainability [32]. AI plays a pivotal role in integrating diverse datasets and facilitating collaborative research efforts. Shared data resources, including crop performance data, weather data, and pest and disease monitoring data, empower stakeholders to make informed decisions and collectively work towards more resilient and sustainable agricultural practices.

2.1.6. Summary

The landscape of AI in agriculture is rich and diverse, with notable developments in technology and user-centered approaches. As the industry advances towards the so-called agriculture 4.0, a comprehensive approach is essential, considering both technical capabilities and human needs. Open-source architectures, MLOps, smart agriculture technologies, and user experience design are key components of this evolving landscape. This article seeks to address this balance, offering the first open-source AI architecture that presents an integrated approach to cover the whole lifecycle of AI models. It facilitates the collaboration of data scientists and ML engineers and provides support for some of the most widely known IoT protocols, hence proving its differential value for the agricultural domain.

2.2. Proposed Architecture

In response to the challenges faced by the agriculture industry to meet the growing global demand for food production while ensuring sustainability and the limitations of available agricultural land, innovative technological solutions have become imperative. This section proposes an architecture that aims to address these challenges by integrating AI in the agricultural landscape. By leveraging open-source technologies and embracing the principles of MLOps, the proposed architecture offers a comprehensive framework to address the complexities associated with the entire lifecycle of AI models, from development to deployment and maintenance.

This section is organized as follows: Section 2.2.1 introduces the fundamental concepts of AI workflows, examines the professional profiles involved, and details the MLOps workflow. Section 2.2.2 presents the architecture proposed in this paper and the technologies employed to shape it. Finally, Sections 2.2.3–2.2.5 delve into each technology used in designing the proposed MLOps architecture.

2.2.1. Fundamentals

To create a new AI model, developers must follow some well-known sequential steps: preprocess and adapt the input data; design, train, and test the model; and finally, deploy the model into a production environment. Even though one could consider that the hardest parts are the first two steps (which usually conform the so-called development stage), it should be noted that, indeed, the last one is crucial, as it implies transitioning the model from a prototype to a real-world product (e.g., production stage).

When starting to experiment with AI models, developers normally use local environments in their local machines. While this method may perform sufficiently well for small, local projects, it starts to be impractical when the models, data, and projects grow.

Specifically, some of the main limitations of using local environments are

- **Data management:** the input dataset for an AI model is stored on the same local machine where the AI model is trained and tested. This makes it impossible to update the dataset in the case of new data being incorporated, as the memory consumption of that local machine would render the infrastructure inoperable.
- **AI model versioning:** tools that allow for the versioning of different trained models are not used, since this process normally involves exchanging considerable amounts of data.
- **Teamwork:** if a team is working on a local machine, the design of pipelines can only be performed and manipulated by one person.
- **Production deployment:** This process often involves additional considerations such as scalability, reliability, security, and monitoring. Furthermore, production deployment can hinder problem solving in a production environment, posing challenges when addressing errors efficiently and promptly.

Therefore, in more complex production environments, additional techniques may become useful to simplify the management, deployment, and monitoring of software developments in a real-world setting. These techniques aim to ensure that the deployment process is efficient and robust, capable of handling the complexities and demands of a live environment. The term used for this culture or set of good practices is Development Operations or DevOps, which combines software architecture with software development, designing software environments that optimize the deployment of applications in high-stress environments. Additionally, in more places, security is taking an important place in DevOps culture, extending to the DevSecOps paradigm [69]. This shows a growing recognition of the key role of security in the development and operations processes, integrating it into every step of the software lifecycle to ensure more resilient and safer applications.

When applied to ML, this paradigm is known as MLOps, primarily concentrating on the deployment of AI models in production.

MLOps exhibits certain differences compared to the original DevOps paradigm, mainly stemming from the unique requirements of AI models, such as model and data versioning and management. These requirements include the need for training with large amounts of data and often require periodic retraining due to widely known issues, such as data drift. Consequently, methodologies and tools that confer agility to the platform must be employed to effectively support these requirements.

With the increase in the complexity of the models to develop, the activities that developers must carry out in each of the stages mentioned before (development and production) also increase in complexity.

In the development stage, developers iterate on several model hyperparameters, evaluate different algorithms, and improve the dataset quality. The result of this stage includes

- **Training Code:** Normally, a Jupyter Notebook [70] is used for experimentations, so it contains the code used for model training, hyperparameter tuning, and evaluation.
- **Trained Model with Artifacts and Versioning:** The developers produce a trained model along with its associated artifacts. These artifacts are versioned to ensure reproducibility and allow for easy tracking of model changes.
- **Stored and versioned Dataset:** The dataset used for training is also versioned to maintain a record of its evolution.

In the production stage, developers focus on turning the Jupyter notebook into a production-ready system using ML pipelines, which bring several advantages, such as versioning, containerization, and collaboration. The steps involved in this stage are

- **ML Pipelines:** These pipelines automate key tasks, including data preprocessing, feature engineering, model (re)training, and evaluation. ML pipelines improve code modularity, facilitate version control, and enable collaboration among team members.
- **Continuous Integration and Delivery (CI/CD) Pipeline:** This pipeline includes steps such as formatting checks, execution of unit tests, and documentation verification. CI/CD ensures code quality, detects errors early, and allows for rapid iteration and deployment.
- **Containerization and Deployment:** In this step, the model, its dependencies, and the ML pipeline are encapsulated into a container, ensuring consistency across different environments. The containerized model is then deployed in the production infrastructure, ready to serve predictions.

Considering the stages in which the AI generic workflow is divided and the complexity that each of them entails two specialized roles come about: data scientist and ML engineer.

- A **Data scientist** is responsible for gathering and preprocessing data, exploring and analyzing datasets, developing and training ML models, and evaluating their performance. Data scientists take the lead in the model development stage, experimenting with different algorithms, hyperparameters, and dataset improvements.
- An **ML engineer** specializes in implementing and operationalizing ML models in production environments. They bridge the gap between data science and software engineering, focusing on deploying, scaling, and maintaining ML systems. ML engineers work on developing robust ML pipelines, optimizing model performance, designing scalable architectures, and ensuring the reliability and scalability of production systems. They define formatting checks, unit tests, and documentation requirements to maintain code quality and ensure successful deployment.

Although they have differentiated tasks, collaboration between data scientists and ML engineers is a key part of an AI model's lifecycle. In terms of knowledge sharing, data scientists and ML engineers must regularly communicate and exchange insights, challenges, and discoveries.

In production stages, ML engineers collaborate with data scientists to deploy the ML pipelines in the production infrastructure. They containerize the model and its dependencies for easy deployment and scalability. Both roles need to work closely to establish

monitoring mechanisms for the deployed model, collecting real-time performance data and iterating on the model or making necessary updates based on the feedback.

The challenge of collaborative work arises from the significant divergence in the technologies utilized by these two roles. The environments and procedural steps followed by each role exhibit notable disparities. This dichotomy often leads to data scientists and ML engineers operating in separate spheres, thereby sharing tools with limited efficiency (Figure 1). Hence, the objective of this endeavor is to create a framework based on an open-source architecture, capable of identifying synergies between these distinct roles. By fostering a more cohesive and collaborative approach, this framework aims to amplify productivity and performance, enabling seamless teamwork (Figure 2).

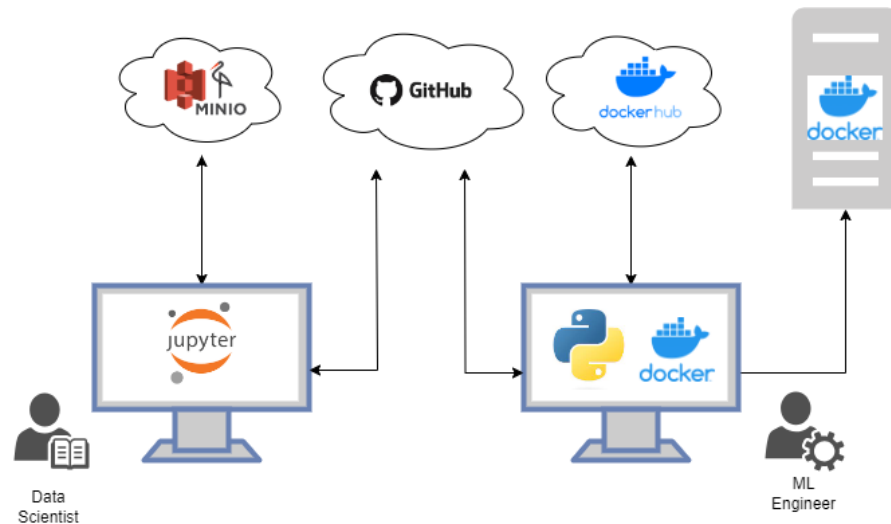


Figure 1. Isolated working method for data scientists and ML engineers.

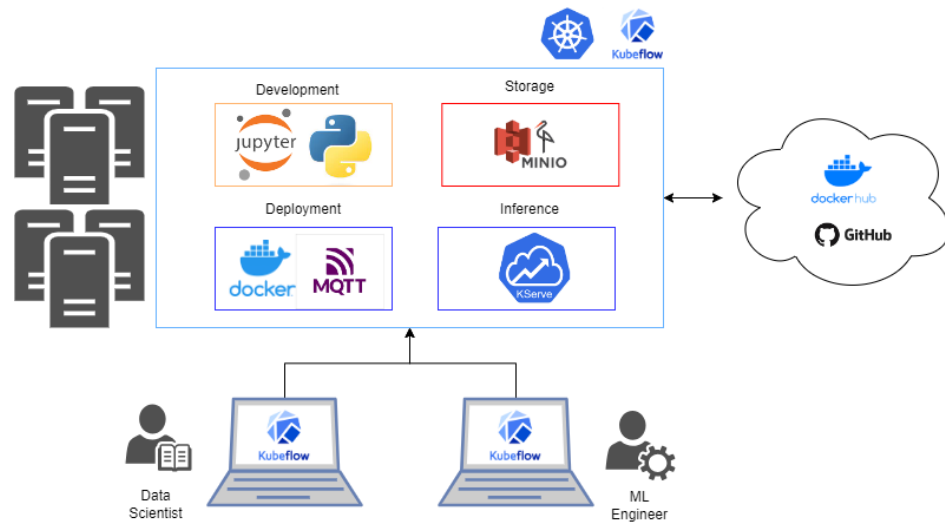


Figure 2. Solution proposed in this paper.

The proposed platform’s high-level architecture, as depicted in Figure 3, highlights three key pipelines: a data management pipeline, a model development pipeline, and a model deployment in production pipeline. This design ensures a comprehensive framework that seamlessly integrates these essential aspects of AI model development and facilitates efficient collaboration between different AI stakeholders.

Data Management is a pivotal process responsible for the comprehensive handling of raw data, from its acquisition to its transformation into a standardized format. This process is composed of several distinct submodules, each with its specialized functions:

- **Data Analysis Submodule:** This module is tasked with assessing and integrating new data into the existing dataset. It maintains a connection with the logs from the *Train, Test, and Fine-tuning Model* submodule, offering insights into training performance and facilitating potential adjustments to the dataset format. As an example, Great Expectations tool [71] is an open-source Python-based library to ensure the reliability of data by asserting certain “expectations” or quality assessments on datasets.
- **Data Versioning Submodule:** Through meticulous documentation, areas for dataset improvement can be identified and the introduction of new dataset versions is registered, ensuring the maintenance of a dynamic dataset. As can be seen in the previous figure, DVC [72] is another open-source tool designed for versioning datasets, model weights, and intermediate files, enabling reproducibility and efficient data pipeline tracking.
- **Data Storage Submodule:** A plethora of technologies is available for data storage, such as MinIO, Amazon S3, PostgreSQL, etc. Notably, this submodule interacts directly with the *Data Preprocessing* submodule within the ML pipeline.

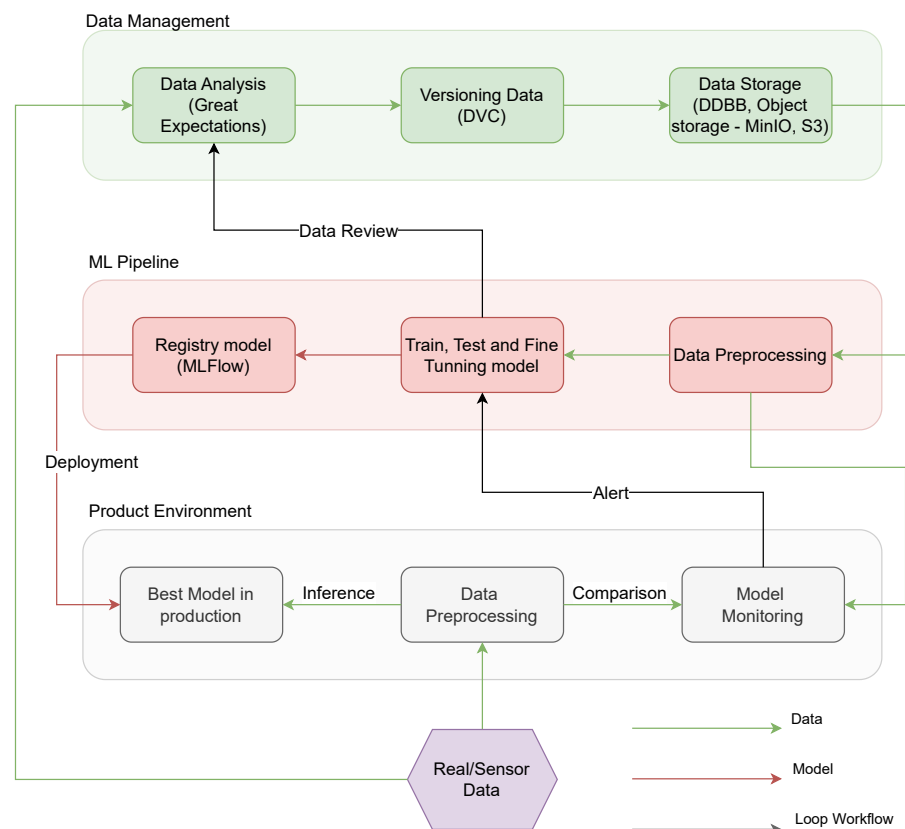


Figure 3. Basic concepts MLOps.

The ML pipeline serves as the foundational element in the proposed architecture, encompassing the programming, training, and testing of the AI model. This pipeline is structured into three distinct submodules:

- **Data Preprocessing:** This submodule retrieves data from the database, undergoes cleaning operations, and restructures them to ensure the model’s optimal training. It is integral to the ML pipeline as it preprocesses data tailored for a specific AI model. Furthermore, the output of this submodule is interconnected with the *Model Monitoring* submodule in the Production Environment, ensuring the reference data remains updated.
- **Train, Test, and Fine-tuning Model:** This submodule ensures that the model’s accuracy, reliability, and performance are aligned with the desired outcomes. If data

restructuring or modifications are required, a comprehensive Data Review is imperative, needing the invocation of the Data Analysis submodule again.

- **Registry Model:** This submodule is dedicated to the preservation of the highest-performing AI models which were trained in the previous submodule. MLFlow tool could be used as an example to register a trained model [73]. As a model registry tool, it centralizes model management, tracks versions, and facilitates lifecycle transitions. The model deemed superior is subsequently deployed in the production environment, bridging the *Registry Model* submodule with the *Best Model in Production* submodule in the Production Environment pipeline.

To complete the lifecycle of both the AI model and data within a tangible system, the deployment into the production environment is crucial.

- **Data Preprocessing in Production:** Upon collection, raw data undergoes preprocessing via the *Data Preprocessing* submodule. These refined data are then fed to the *Model Monitoring* submodule, where its coherence is meticulously analyzed.
- **Best Model in production:** The processed data from the previous component is then channeled into the submodule dedicated to model inference.
- **Model Monitoring and Alerts:** The output data are further integrated into the *Model Monitoring* submodule for an in-depth analysis. Should the data received from the sensors and the model's output diverge beyond the acceptable variance, the system triggers an alert, highlighting the potential need for training a novel model.

2.2.2. Technological Implementation

As outlined in the previous section, this study presents a framework detailing the key functions an MLOps platform should include to overcome limitations commonly found in sequentially based programs. This section offers a detailed review of open-source tools that facilitate the effective design and implementation of this platform, particularly for its use in agriculture.

Open-source technologies offer several key advantages for research and development. Firstly, they promote transparency, allowing researchers to access and inspect the source code, ensuring a high level of trust and credibility in the results. Secondly, they foster collaboration by encouraging a global community of developers to contribute, resulting in continuous improvement and innovation. Additionally, open-source tools are often cost-effective, making them accessible to a wider range of researchers and organizations. Lastly, they provide flexibility and customization options, enabling personalized solutions to specific research challenges. These benefits collectively enhance the efficiency, reliability, and impact of research projects, since the only cost associated to the usage of the proposed architecture would be associated to the underlying hardware infrastructure. For illustrative purposes, the work presented in here is supported by a server cluster with a total of 180 cores of varied types and 300 GB of RAM memory. Additionally, this cluster is equipped with two GPUs, specifically Nvidia T4 and A100.

The utilization of this architecture in agricultural settings is justified for several reasons. First, these systems handle a large volume of inputs and outputs, generated by sensors in various agricultural tools. Due to this complexity, it is essential to have architectures that enable agile deployment and deactivation of tools, a function efficiently fulfilled by the proposed architecture, as will be explained further in the document.

Secondly, when used in agricultural contexts, these systems often rely on the Internet of Things (IoT) paradigm [74], which normally involves a network of interconnected physical devices, each equipped with sensors, software, and connectivity options. These features not only facilitate data exchange but also enable interactions with the environment. In this context, the primary aim of the architecture is to seamlessly integrate these IoT agents in agricultural settings with existing production models. To achieve this, the platform uses well-known IoT protocols, such as MQTT (designed for device-to-device communication) and HTTP (used for data packet exchange in cloud environments). These protocols also contribute to the system's communication redundancy and agility, ensuring that, in the

event of a channel failure, devices can continue to communicate via alternative routes. This strategic integration allows data scientists and ML engineers to develop or fine-tune models more efficiently, as previously outlined in Section 1.

The proposed architecture is built on an on-premise Kubernetes platform, which is a leading tool for container orchestration. This foundation provides strong stability and resilience for the included applications. Using Kubernetes' features allows the platform to scale, easily deploy applications, automate management, and recover from faults. Kubernetes serves as the fundamental building block of this solution, guiding the deployment and management of application containers. These containers, which host the different components of our architecture, are light and portable, ensuring consistency across various deployment environments and simplifying packaging processes. The architecture is illustrated in Figure 4 which is composed of DevOps-based tools, such as Kubernetes, MQTT, CamelK, and state-of-the-art tools which support AI tasks, such as Kubeflow, MLFlow, and Kserve.

It clarifies the key components and how they interact for maximum efficiency. As a result, the overall architecture, supported by Kubernetes, is designed to work smoothly with the open-source tools. This approach aligns with the concept displayed in Figure 3, with later sections delving into the specific role of each open-source tool within the MLOps framework.

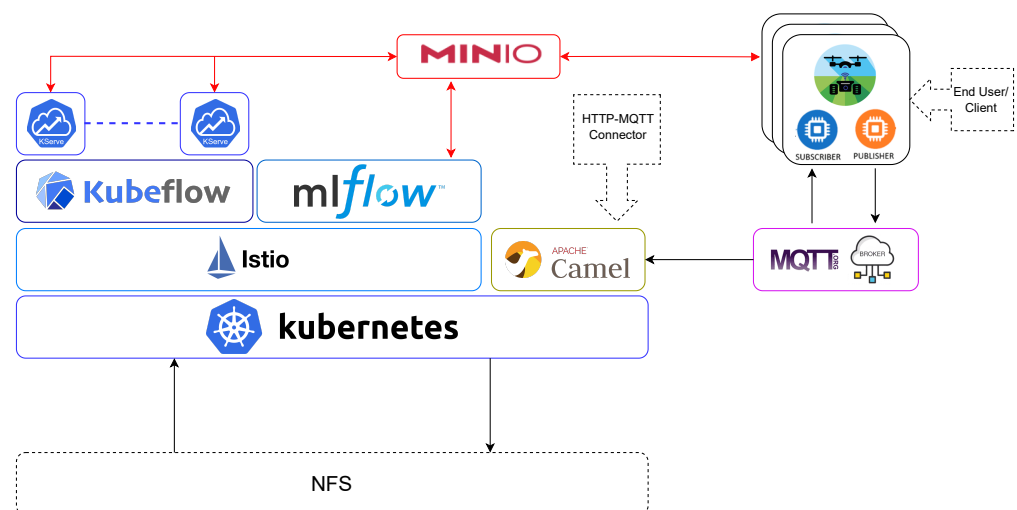


Figure 4. Specialized MLOps Architecture for Agricultural Applications.

It is important to note that the code and setup instructions for the previously described architecture are openly accessible through the official repository of the FlexiGroBots European project [75], which aims to build a platform for intelligent automation of precision agriculture operations. For public interaction and model inference, a dedicated website is available at [76]. Moreover, the platform offers a centralized workspace exclusively for data scientists and ML engineers involved in the FlexiGroBots European project [77]. This exclusive access to this common entrypoint allows for seamless AI model sharing, development, and collaboration among them. Subsequent subsections will delve deeper into the three pipelines/workflows previously outlined in earlier sections and depicted in Figure 3, providing a comprehensive overview of the architecture.

2.2.3. Data Management

The platform described in this document starts with a module called “Data Management”. As shown in Figure 3, this module’s scope encompasses the storage, versioning, and provision of data access.

In the development and deployment of AI models, efficient storage and retrieval of large datasets and models is crucial. While Kubernetes offers native storage solutions, they

come with limitations, particularly in terms of flexibility and mobility. To address these constraints, a specialized tool that enhances storage flexibility has been employed, allowing for dynamic adjustments as required by the system. To address this, the Kubernetes Network File System (NFS) Subdir External Provisioner is used. This tool allows for the creation of volumes by every component in the cluster using an NFS server and makes storage more flexible by dynamically setting up subdirectories on that server.

In light of this dynamic storage management approach, MinIO is identified as the ideal open-source tool for handling data and versioning. Known for its high performance, MinIO is a distributed object storage system designed for handling large amounts of data and providing fast data access. It offers features such as scalability, fault tolerance, and specialized performance for object storage tasks. When combined with Kubeflow (which will be explained in the following subsection), MinIO acts as a robust storage layer capable of archiving models and datasets. Additionally, MinIO is user-friendly due to its easy-to-use interface and straightforward setup process, making it easy for users to quickly implement MinIO for their storage needs.

The MinIO platform addresses the Data Management pipeline by establishing communication with other submodules of the platform via HTTP. Furthermore, it is capable of storing data using the bucket storage format. In addition, MinIO provides a system for versioning the buckets used for model training.

2.2.4. ML Pipeline

In order to create the ML pipeline depicted in Figure 3 and leverage Kubernetes [78] technology as the foundation of the platform, a Kubeflow solution [79] was configured and deployed. Kubeflow provides a comprehensive suite of tools that enable data scientists and ML engineers to develop, train, deploy, and monitor ML models effectively. When installing Kubeflow, Istio [80] is also set up. Istio manages microservices, enhancing security and directing traffic between tools in the cluster. The adoption of this tool arises from the need to secure both the inputs and outputs of a system when utilizing open-source technologies, as illustrated in [81]. Since these tools typically lack a dedicated company to address bugs, the responsibility falls on the community. This underscores the importance of employing encrypted and secure systems, such as those facilitated by Istio. Thanks to this service mesh, we can implement TLS encryption in both external communications to the architecture and internal communications between the infrastructure's internal modules, allowing us to secure both the internal and external connections with the architecture. This ensures smooth and secure communication between these tools which will be used by data scientists and ML engineers. By leveraging Kubeflow's features, users can focus on building robust ML models while minimizing their involvement in underlying infrastructure complexities.

Kubeflow encompasses various tools and components tailored to different stages of the ML workflow:

- For data preparation and task orchestration for training, Kubeflow Pipelines [82] offers a visual interface that facilitates the design and execution of data processing pipelines or automated model training.
- In terms of experimentation, development, and model training, Kubeflow integrates with popular ML frameworks such as TensorFlow [83], Pytorch [84], and scikit-learn [85]. Experiments and developments can be carried out by using its integrated Jupyter Notebook implementation. Jupyter Notebook is a web-based tool for creating, sharing, and executing files containing live code, visualizations, and explanatory text. These Jupyter Notebooks are managed as Docker [86] containers by Kubeflow, easing their deployment and versioning. Kubeflow provides tools such as Katib [87] for studying model hyperparameters during training.
- Regarding model serving and deployment, Kubeflow supports the deployment of models as scalable, production-ready services using Kubernetes [78] serving frameworks such as Tensorflow Serving [88] or Seldon Core [89] along with KServe [90].

- As for monitoring and observability, users can monitor model performance, track key metrics, and set up alerts to ensure the models function as intended.

As shown in Figure 5, a Machine Learning Operations (MLOps) process was set up using pipelines. This involves various stages of developing, using machine learning models in a Kubernetes environment, with Kubeflow playing a major role. This specific process is focused on agricultural applications, where it is used by agronomists.

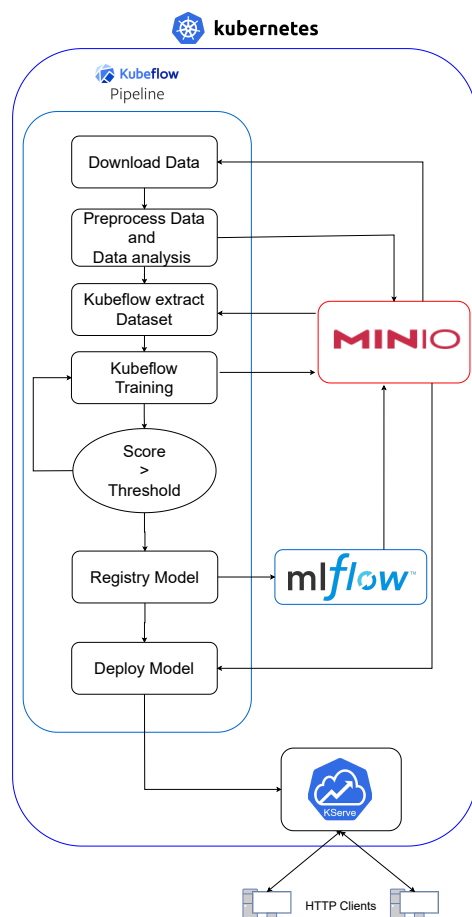


Figure 5. Kubeflow architecture.

The first step is collecting data from different sources. These data are then preprocessed, which includes cleaning, normalizing, and extracting features to make it ready for training models.

Following the preprocessing, the Kubeflow toolkit comes into play. Renowned for its robustness in managing machine learning workflows in Kubernetes, Kubeflow handles the dataset extraction. This step is pivotal in structuring the data into a format that is optimized for subsequent machine learning processes. Additionally, after the data processing, data analysis techniques for anomaly detection are employed, as demonstrated in the work cited [91], used to ensure the quality of time series-based datasets. This is to ensure the quality and balance of the data when training the models.

Next, successful models are registered. This involves keeping track of different versions and their details. Then, these models are deployed and available for practical use, being, in this step, a key part of the project focused on agricultural engineering.

As a result of this pipeline, several models have been integrated into an easy-to-use web platform [76] and are also accessible via the Kserve API-REST, allowing users to obtain information directly.

Agronomists, the main users of this system, interact with the models via a web interface. They input real-world agricultural data and obtain back predictions for analysis.

This use of the models shows how they can provide helpful insights for agricultural engineering.

In conclusion, this MLOps process combines data processing, machine learning, and workflow management in Kubernetes. It is designed to meet the needs of agronomists, showing how machine learning can be applied effectively in agriculture.

2.2.5. Production Environment

The architecture discussed in this paper has been designed to support the use of AI models after their development and deployment. This focus is particularly on key elements of MLOps, such as sensor interaction with the system and AI model deployment. It is important to note that sensors usually communicate via the HTTP protocol. Therefore, these types of data can be easily processed using KServe's API.

However, in the agricultural setting, sensors often come in arrays that generate large data streams. These arrays are managed using another protocol called MQTT, which is designed for real-time applications and is efficient in terms of bandwidth and computational resources. MQTT uses a publish–subscribe system, ensuring efficient and direct data exchange between data sources from sensors and AI models.

To facilitate this, the architecture has been adapted to allow KServe to interact with the MQTT protocol. Two components have been added to the platform, enabling models hosted on KServe to collect sensor data, perform inferences, and return the results via MQTT or HTTP, depending on the project's needs.

To access MQTT messages in the broker, CamelK [92] is used, acting as an HTTP–MQTT connector. This allows data to be served to KServe modules, which can then redirect the information via HTTP or MQTT as needed.

Given the nature of the agricultural field, another important challenge was anticipating and addressing potential communication and data management challenges. To ensure robust and efficient data handling, the presented architecture incorporates two methodologies. Firstly, asynchronous-type protocols, such as MQTT, have been used. This strategy serves a dual purpose: it not only distributes data loads efficiently across communication channels, but also fortifies the system's overall resilience. In this protocol, when a broker receives a topic, it retains the information until a subsequent request is made, assuring that, upon system connection, the necessary data are dispatched promptly. Additionally, MQTT provides a unified communication channel for all sensors in the agricultural ecosystem, from drones to autonomous tractors, streamlining automated management processes. To complement this, as mentioned in Section 2.2.3, a storage system known as a data lake has been installed. Situated within the same cluster, this storage solution guarantees quick and direct access to crucial data for all managing components.

3. Results

The following section presents the results and findings obtained from the evaluation of the platform presented in this paper.

Firstly, we have demonstrated practical applications of the graphical user interface, specifically designed to facilitate the accessibility of inference systems for individuals with non-technical backgrounds.

Secondly, a thorough analysis was conducted to assess the performance, usability, and effectiveness of the proposed architecture. As part of this process, potential users of the platform have been surveyed regarding the usage of this solution. This evaluation aimed to validate the platform's capabilities and its potential impact in addressing the challenges outlined in Section 1. The results provide valuable insights into the platform's qualitative assessments, comparative analysis against existing solutions, scalability and efficiency, as well as real-world use cases. Additionally, limitations identified during the evaluation are discussed, and potential future directions for further improvement are suggested.

Finally, the platform's evaluation of its position within the broader scope of MLOps must be considered. This is particularly important as MLOps is rapidly evolving and

becoming more prominent in fields such as agriculture. As a recent research study shows, MLOps primarily aims to streamline AI operations, improve collaboration, and facilitate the transition from AI model design to deployment [93]. The survey data, especially from AI professionals, supports the platform's ability to effectively bridge the gap between data scientists and ML engineers, a sentiment also highlighted in [94].

The section is organized as follows: first, Section 3.1 presents the designed GUI for end-users to interact in a visual and simple way with inference systems. Second, Section 3.2 presents a summary of how different stakeholders can use the proposed architecture; then, Section 3.3 outlines the contents of the survey conducted, together with the insights extracted from them.

3.1. Web Platform

In this work, a Graphical User Interface (GUI) (Figure 6) has been developed for individuals who lack a technical background in making queries to REST APIs or in deploying AI models. Therefore, in the GUI, four inference models have been deployed for use by Agricultural Engineers. In addition, this GUI is public [76] via internet.

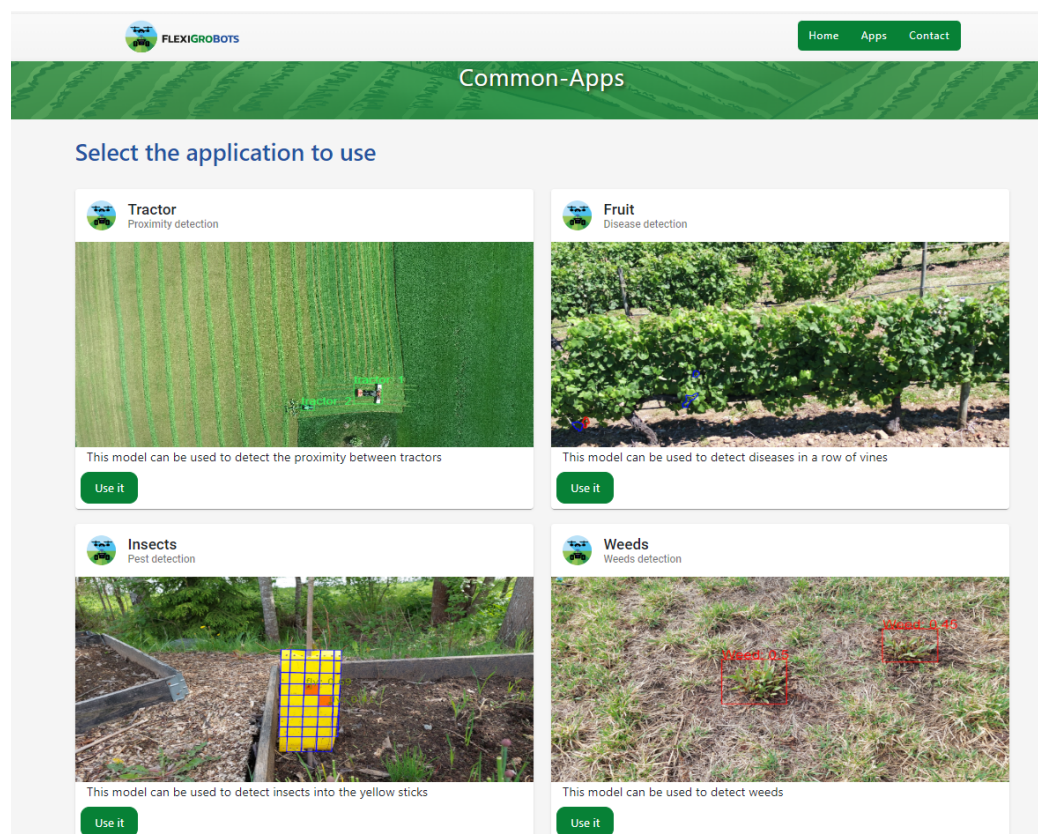


Figure 6. Web platform with agriculture tools.

In this study, four distinct Artificial Intelligence models have been deployed for specific agricultural applications. These models represent cutting-edge integrations of AI in the realm of agriculture, addressing diverse challenges faced by the industry.

Firstly, a model dedicated to the detection of tractors utilizing computer vision techniques has been implemented (top left corner in Figure 6). This model exemplifies the application of image recognition technologies in agricultural settings, enabling enhanced monitoring and management of farming equipment.

Secondly, the project's focus extends to the precise detection of Botrytis, a significant fungal disease affecting various crops (top right corner in Figure 6). Leveraging AI, a model that facilitates targeted spraying through advanced detection methods has been developed.

This approach allows for precise application of fungicides, optimizing resource use and minimizing environmental impact.

Thirdly, pest management is addressed via the deployment of an AI model capable of detecting and counting insects in traps (bottom left corner in Figure 6). This model facilitates the monitoring process, providing accurate, real-time data that is crucial for effective pest control strategies.

Finally, a model for the detection and quantification of weeds has been introduced (bottom right corner in Figure 6). This model aids in the identification of unwanted flora, a key task for crop management and yield optimization. By accurately recognizing and counting weed species, this model supports more effective and environmentally conscious weed control practices.

Collectively, these AI models demonstrate the potential of Artificial Intelligence to revolutionize various aspects of agricultural operations, offering innovative solutions to longstanding challenges in the field.

3.2. How to Use the Platform

As introduced in Section 2.2, one of the main motivations of the work presented in this paper is to provide a comprehensive architecture that eases the collaboration between two clear and distinct professional profiles that have arisen in these last years: data scientists and ML engineers. Without this architecture, the collaboration between these roles is not efficient, which results in limited job parallelization. Typically, the ML engineer waits for the data scientist to finalize the model's preparation and tuning. Then, the ML engineer must build the production infrastructure from scratch. If issues arise, adjustments by the data scientist are followed by reintegration by the ML engineer. These development and deployment processes are lengthier due to the required coordination between both roles, as depicted in Figure 7a.

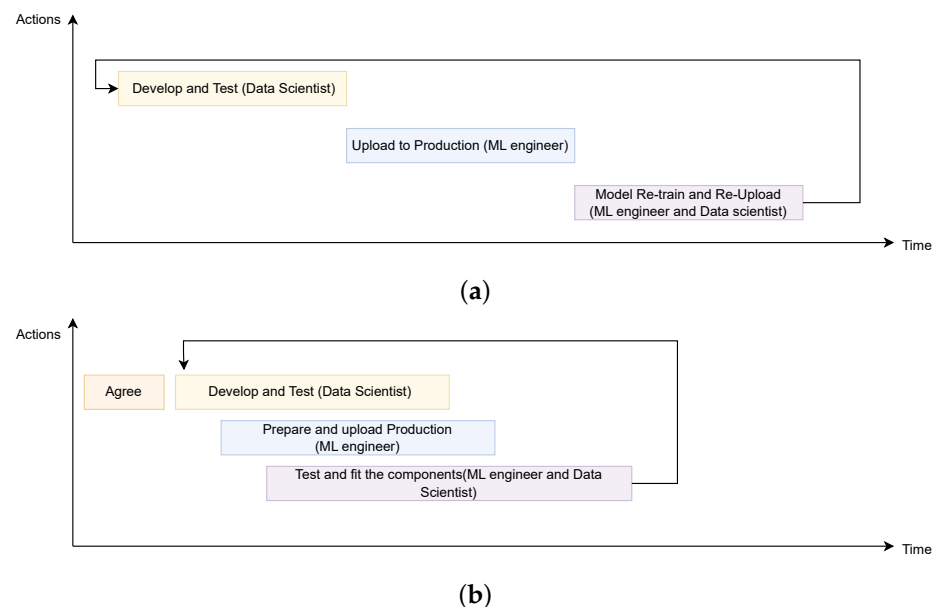


Figure 7. Comparative using standard and MLOps method. (a) Development and deployment without MLOps platform. (b) Development and deployment with MLOps platform.

The architecture proposed in this paper (Figure 4) provides an ideal solution for this problem, empowering each role to excel in their respective stages. The proposed method delivers results in significantly shorter times compared to the traditional development methodology described above. In this process, collaboration between the data scientist and the ML engineer is crucial, defining the AI solution before development begins. Figure 7b il-

illustrates an initial “agreement” step, outlining the pipeline components and their respective inputs and outputs.

Once the data scientist is clear about their role and understands the ML engineer’s procedure, they can start designing AI solutions using Jupyter notebooks provided by Kubeflow. They also have the option to use Katib for hyperparameter tuning studies. Concurrently, to expedite the process, the ML engineer begins designing pipeline components, such as data collection and AI model deployment in KServe. Since this stage can also be executed with Kubeflow’s Jupyter notebooks, the ML engineer can integrate preliminary versions of the model into the full pipeline.

Upon the data scientist’s model design completion, the ML engineer integrates the model, gathering the necessary weights for its pipeline implementation. During this phase, the ML engineer can also adjust the model’s input protocol, incorporating data from HTTP or MQTT. Thus, the ML engineer prepares the AI model for actual production deployment.

The outcome is a comprehensive and efficient pipeline covering the entire process of development, testing, and deployment of an AI model.

Finally, the proposed architecture allows data scientists and ML engineers to track the performance of the deployed model in real time by using the same platform. They can collect relevant metrics, monitor the model’s behavior, and make iterative improvements based on the feedback received. This feedback loop ensures continuous enhancement and adaptation of the model to changing requirements.

3.3. Conducted Interviews

To assess the utility and effectiveness of the proposed platform, interviews were conducted with six data scientists and five ML engineers who are involved in diverse agricultural field activities. The interviewees possessed mid-level seniority and have been working in the machine learning field for the last 3–5 years, giving a balanced perspective of experience and current industry practices to the study. Therefore, it has been determined that the responses provided by the respondents carry equal weight, as they all possess the same level of expertise. In Section 3.3.1, the contents and structure of the survey are presented. Afterwards, in Section 3.3.2, the results obtained during the surveys are presented in a pointed format from 1 to 5. Additionally, metrics such as the mean and standard deviation have also been extracted to statistically comprehend the results.

3.3.1. Contents and Structure of the Survey

To evaluate the usability and adaptability of the platform in general, the surveyed professionals were invited to implement the platform in collaborative projects, aiming to understand its adaptability and performance in a real-world setting. One of the first aspects examined was the learning curve, measuring the time it took for professionals to become comfortable using the platform’s tools and architecture in an actual project.

The choice to center this evaluation on interviews with AI professionals arises from the intention to ensure that the proposed platform addresses tangible challenges in real-world settings. The insights gained from these interviews offer a comprehensive view of the platform’s capabilities. This emphasizes its effectiveness in closing communication and operational gaps between data scientists and ML engineers [93]. This human-centric evaluation methodology highlights the commitment to utility and adaptability.

Additionally, special attention was given to understanding how the platform would impact the team dynamics. Factors such as collaboration and efficient task division were evaluated to determine whether the platform facilitates or complicates these processes. This is crucial, as any tool, no matter how advanced, must effectively integrate into a team’s workflow to be genuinely useful.

Another area of focus was the interaction between different professional profiles. The aim was to assess whether the platform simplifies communication and collaboration between data scientists and ML engineers, who often need to coordinate closely but from different technical perspectives.

The study also explored the transition of projects from development to production. This is key to understanding the platform's versatility and its applicability at different stages of a project's lifecycle.

In terms of data management, the user experience with the platform's storage system (MinIO) was examined, especially concerning data handling and sharing. Lastly, the ease with which knowledge of the platform can be transferred to new users was evaluated, an aspect for the long-term sustainability of projects.

In summary, the evaluation aimed to show how well the platform meets the needs and solves the challenges faced by both data scientists and ML engineers.

3.3.2. Metrics

For a better understanding of the results and system distribution, in addition to the theoretical explanation, each interviewee was asked to provide a rating indicating the level of difficulty, ranging from 1 (most negative) to 5 (most positive). Each aspect mentioned in the previous section is considered one of the evaluated criteria.

These ratings have been accumulated and organized into tables, where the "criterion" column displays the corresponding question, and the remaining table values store the accumulative scores given by the respondents.

Furthermore, to provide more context and understand the statistical variation in these values, it has been decided to calculate the mean and standard deviation. Additionally, it has been assumed that the distribution is normal and confidence intervals have been constructed around the mean, where 70% aligns with the criterion [$\bar{x} - z * \sigma$, $\bar{x} + z * \sigma$].

In Equations (1) and (2), the value n represents the number of elements in the dataset, and x_i represents the individual values in the dataset.

$$\bar{x} = \frac{1}{n} \sum_{i=1}^n x_i \quad (1)$$

$$\sigma = \sqrt{\frac{1}{n} \sum_{i=1}^n (x_i - \bar{x})^2} \quad (2)$$

Following Table 1, the majority of the scores assigned to the data scientist team are above 3. This suggests that the infrastructure is accessible to data scientist profiles. As observed in Table 2, the statistics for all criteria, specifically the mean, are above 4 out of 5, with a standard deviation of ± 1 . This demonstrates that the evaluated individuals feel comfortable and at ease with all the assessed criteria.

Table 1. Data scientist evaluations.

Criterion	Data Scientists				
	1	2	3	4	5
Learning Curve	0	0	1	3	2
Teamwork	0	0	0	3	3
Interaction with Other Profiles	0	0	1	3	2
Production	0	0	1	2	3
Data Management	0	0	2	2	2
Handover	0	0	0	2	4

Table 2. Data scientist statistics.

Data Scientists	\bar{x}	σ	$\bar{x} - z * \sigma$	$\bar{x} + z * \sigma$
Learning Curve	4.16	0.68	3.47	4.85
Teamwork	4.5	0.50	4.00	5.00
Interaction with Other Profiles	4.16	0.68	3.47	4.85
Production	4.33	0.74	3.58	5.00
Data Management	4.0	0.81	3.18	4.81
Handover	4.66	0.47	4.19	5.12

In Table 3, the data reveals a noteworthy trend, with the majority of the scores awarded to the Machine Learning engineer team surpassing the 3-point mark, affirming the accessibility of the infrastructure for professionals specializing in Machine Learning. Based on the statistical analysis presented in Table 4, each criterion, particularly the mean scores, consistently register above the 4-point threshold out of 5. Accompanied by a standard deviation range of ± 1 , these findings affirm the participants' pronounced comfort and proficiency across the spectrum of evaluated criteria.

Table 3. Machine Learning engineers' evaluations.

Criterion	Machine Learning Engineers				
	1	2	3	4	5
Learning Curve	0	0	0	1	4
Teamwork	0	0	0	3	2
Interaction with Other Profiles	0	0	1	4	0
Production	0	0	0	5	0
Data Management	0	0	2	1	2
Handover	0	0	1	2	2

Table 4. Machine Learning engineers' statistics.

Machine Learning Engineers	\bar{x}	σ	$\bar{x} - z * \sigma$	$\bar{x} + z * \sigma$
Learning Curve	4.80	0.40	4.40	5.20
Teamwork	4.40	0.49	3.91	4.89
Interaction with Other Profiles	3.80	0.40	3.40	4.20
Production	4.37	0.48	3.89	4.85
Data Management	4.00	0.89	3.10	4.89
Handover	4.20	0.74	3.45	4.94

4. Discussion

A summary of the key findings and conclusions extracted from the surveys conducted with the professionals aforementioned is provided in Table 5. In the following, the contents of the table are going to be analyzed in detail.

Both data scientists and ML engineers encountered a somehow steep initial learning curve when starting with MLOps tools, particularly with Kubeflow. However, this challenge was generally overcome after an initial period, leading to a more intuitive and user-friendly experience with the platform's tools. It is important to mention that prior knowledge in related technologies, such as Kubernetes, was cited as beneficial for easing the learning process.

Table 5. Summary of the answers to the survey.

Aspect	Data Scientists	Machine Learning Engineers
Learning Curve	Initial learning curve is steep, especially with Kubeflow. Becomes intuitive after initial phase.	Initial learning curve with Kubeflow, eased by prior knowledge in Kubernetes.
Teamwork	Task division is challenging due to diverse technologies. Some level of simultaneous collaboration is possible.	Facilitates collaboration but requires well-organized team and clear methodologies.
Interaction with Other Profiles	Facilitates interaction but requires initial organization.	Smooth interaction with Data Scientists, allows for parallel work streams.
Production	Simplified by tools such as MLFlow and KServe.	Considered to streamline the process.
Data Management	Centralized data storage in MinIO is beneficial.	Centralized data storage in MinIO allows for easy sharing and customization.
Handover and Data Sharing	Time-consuming if the other party is not familiar with the technologies. Otherwise, straightforward.	Straightforward if both parties are familiar with the platform.

In terms of collaboration and teamwork, the platform was universally seen as a facilitator. However, this benefit was not without its conditions. Both roles emphasized the need for a well-organized team and clear methodologies to fully leverage the platform's capabilities. The structure of the platform allows for the division of tasks and parallel work streams, which can speed up project timelines and make the workflow more efficient. However, this requires the team to be well coordinated and possibly adhere to agile methodologies.

Data management was another common area of agreement. The centralized data storage capabilities provided by MinIO were highly valued by both data scientists and ML engineers. This feature not only simplifies data management but also enhances collaboration by ensuring that all team members have access to the latest versions of datasets.

When it comes to the handover of projects and the sharing of resources, the experiences were generally positive. The platform's structure and the use of code for defining pipelines make it easier to transfer work between team members. However, the time required for handover could increase if the incoming member is not familiar with the platform's technologies. Despite this, once the handover is complete, sharing data, models, and pipelines within the platform is relatively straightforward.

In conclusion, both data scientists and ML engineers find significant value in using MLOps tools, despite facing different challenges and benefits. The common themes that emerged from the study include the importance of documentation, the initial learning curve, and the benefits of centralized data storage.

5. Conclusions and Future Lines

In this study, the development of AI models and their deployment in real productive systems using traditional methods has proven to be complex. To address this challenge, two solutions have been proposed. First, a development method facilitates the natural development and deployment of models. Second, an MLOps-oriented architecture designed for the agricultural sector incorporates specific protocols (MQTT) and open-source technologies to develop and execute AI models, allowing for seamless communication with agricultural devices.

This conclusions section summarizes the primary benefits and insights of the platform, highlighting its significance and potential impact.

- The platform provides access to high-performance infrastructure, enabling efficient utilization of computational power. This capability empowers users to tackle complex tasks and handle large datasets without being constrained by their local hardware limitations.
- The platform serves as a centralized entrypoint which enhances communication and collaboration among data scientists, ML engineers, and other stakeholders, such as domain experts and project managers. This unified platform facilitates the seamless

sharing of code, documentation, and project updates, promoting efficient code review, feedback, and iteration cycles.

- The platform takes care of role and user management, eliminating the need for users to handle multiple credentials for various tools and services. This centralized approach simplifies user onboarding, access control, and overall security.
- The platform seamlessly manages the deployment and monitoring of ML models. This allows for timely issue detection and facilitates proactive maintenance and improvements.
- The platform incorporates centralized storage, streamlining the collaboration and sharing of datasets and ML artifacts. This centralized storage enables users to efficiently manage and access shared data, thereby accelerating the development process.
- Feedback from data scientists and ML engineers who have used the platform indicates its value for enhancing collaboration. However, they also mention that the initial steps can be somewhat challenging for new users. Despite this, they also highlight the need for structured and common methodologies to better organize resources within the platform and streamline day-to-day operations.
- The platform is tailored for agriculture, efficiently connecting with numerous sensors, compatible with MQTT, and built on adaptable open-source technologies, making it an ideal choice for diverse farming needs. The platform has specifically been designed considering the farmer's needs, building it in such a way that the adoption of this architecture would only require a minimum hardware infrastructure to deploy the system.
- While this platform was originally developed for an agricultural environment, its modular and adaptable architecture allows for its application in diverse fields. With appropriate modifications, the platform could be extrapolated to cater to the specific needs and challenges of other sectors.

As with any evolving technology, the platform presents opportunities for refinement and expansion. Considering the shifting nature of a brand-new paradigm such as MLOps and the feedback gathered from the user community, certain areas have been identified for further exploration and enhancement.

- **Improving User Onboarding:** Given the feedback regarding the initial challenges faced by new users, it becomes imperative to make the learning curve smoother. Interactive tutorials, context-sensitive help sections, and even AI-guided walkthroughs are potential solutions to better assist new users as they navigate and familiarize themselves with the platform.
- **Model Monitoring:** While the platform handles the entire lifecycle of ML models proficiently, there is a clear gap in terms of continuous model monitoring in a productive environment. The immediate next step, thus, is integrating well-established tools such as Grafana [95] and Prometheus [96], since they could provide real-time insights, performance metrics, and anomaly detection for deployed models. Although the platform efficiently manages the lifecycle of ML models, it lacks continuous model monitoring in a production environment. To address this, integrating tools like Grafana [95] and Prometheus [96] is essential for offering real-time insights and performance metrics, including anomaly detection for deployed models.
- **More experimentation:** The aim is to engage more data scientists and ML engineers on the platform to maximize its memory and computing capacities. This entails encouraging experts to perform resource-intensive tasks, challenging the system's ability to handle extensive data processing and complex algorithms. By pushing the platform to its limits, it can showcase its scalability and effectiveness for demanding data science and machine learning tasks.

Author Contributions: Conceptualization, A.C.C.-P., Y.L. and R.L.; methodology, R.L.; software, A.C.C.-P. and Y.L.; validation, A.C.C.-P., Y.L. and R.L.; data curation, A.C.C.-P. and Y.L.; writing—original draft preparation, A.C.C.-P., Y.L. and R.L.; writing—review and editing, A.C.C.-P., Y.L. and R.L.; supervision, R.L. All authors have read and agreed to the published version of the manuscript.

Funding: This work has been carried out in the scope of the H2020 FlexiGroBots project, which has been funded by the European Commission in the scope of its H2020 programme (contract number 101017111, <https://flexigroBots-h2020.eu/>, accessed on 1 January 2024).

Data Availability Statement: Code and setup instructions can be found at [75]. For public interaction and model inference, a dedicated website is available at [76].

Acknowledgments: The authors would like to acknowledge the valuable help and contributions from all partners of the FlexiGroBots consortium, as well as the data scientists and ML engineers that participated in the survey.

Conflicts of Interest: The authors declare no conflicts of interest.

Abbreviations

The following abbreviations are used in this manuscript:

AI	Artificial Intelligence
CD	Continuous Delivery
CI	Continuous Integration
DL	Deep Learning
HTTP	Hypertext Transfer Protocol
IoT	Internet of Things
MLOps	Machine Learning Operations
ML	Machine Learning
MQTT	Message Queuing Telemetry Transport
NFS	Network File System
UAV	Unmanned Aerial Vehicles
UGV	Unmanned Ground Vehicles
UN	United Nations
WRI	World Resources Institute

References

1. Ranganathan, J.; Waite, R.; Searchinger, T.; Hanson, C. How to Sustainably Feed 10 Billion People by 2050, in 21 Charts. 2018. Available online: <https://www.wri.org/insights/how-sustainably-feed-10-billion-people-2050-21-charts> (accessed on 12 December 2023).
2. De Clercq, M.; Vats, A.; Biel, A. Agriculture 4.0: The future of farming technology. In Proceedings of the World Government Summit, Dubai, United Arab Emirates, 29 December 2018; pp. 11–13.
3. Wakchaure, M.; Patle, B.; Mahindrakar, A. Application of AI Techniques and Robotics in Agriculture: A Review. *Artif. Intell. Life Sci.* **2023**, *3*, 100057. [CrossRef]
4. Eli-Chukwu, N.C. Applications of Artificial Intelligence in agriculture: A review. *Eng. Technol. Appl. Sci. Res.* **2019**, *9*, 4377–4383. [CrossRef]
5. Elbasi, E.; Mostafa, N.; AlArnaout, Z.; Zreikat, A.I.; Cina, E.; Varghese, G.; Shdefat, A.; Topcu, A.E.; Abdelbaki, W.; Mathew, S. Artificial intelligence technology in the agricultural sector: A systematic literature review. *IEEE Access* **2022**, *11*, 171–202. [CrossRef]
6. Shankar, P.; Werner, N.; Selinger, S.; Janssen, O. Artificial Intelligence driven crop protection optimization for sustainable agriculture. In Proceedings of the 2020 IEEE/ITU International Conference on Artificial Intelligence for Good (AI4G), Geneva, Switzerland, 21–25 September 2020; pp. 1–6.
7. Allmendinger, A.; Spaeth, M.; Saile, M.; Peteinatos, G.G.; Gerhards, R. Precision chemical weed management strategies: A review and a design of a new CNN-based modular spot sprayer. *Agronomy* **2022**, *12*, 1620. [CrossRef]
8. Saxena, R.; Joshi, A.; Joshi, S.; Borkotoky, S.; Singh, K.; Rai, P.K.; Mueed, Z.; Sharma, R. The role of artificial Intelligence strategies to mitigate abiotic stress and climate change in crop production. In *Visualization Techniques for Climate Change with Machine Learning and Artificial Intelligence*; Elsevier: Amsterdam, The Netherlands, 2023; pp. 273–293.
9. Visentin, F.; Cremasco, S.; Sozzi, M.; Signorini, L.; Signorini, M.; Marinello, F.; Muradore, R. A mixed-autonomous robotic platform for intra-row and inter-row weed removal for precision agriculture. *Comput. Electron. Agric.* **2023**, *214*, 108270. [CrossRef]

10. Conejero, M.; Montes, H.; Andujar, D.; Bengochea-Guevara, J.; Rodríguez, E.; Ribeiro, A. Collaborative smart-robot for yield mapping and harvesting assistance. In *Precision Agriculture'23*; Wageningen Academic: Wageningen, The Netherlands, 2023; pp. 1067–1074.
11. Talib, M.A.; Majzoub, S.; Nasir, Q.; Jamal, D. A systematic literature review on hardware implementation of Artificial Intelligence algorithms. *J. Supercomput.* **2021**, *77*, 1897–1938. [[CrossRef](#)]
12. Sufi, F. Algorithms in low-code-no-code for research applications: A practical review. *Algorithms* **2023**, *16*, 108. [[CrossRef](#)]
13. Dogan, M.E.; Goru Dogan, T.; Bozkurt, A. The use of Artificial Intelligence (AI) in online learning and distance education processes: A systematic review of empirical studies. *Appl. Sci.* **2023**, *13*, 3056. [[CrossRef](#)]
14. Mhlanga, D. Industry 4.0 in finance: The impact of Artificial Intelligence (ai) on digital financial inclusion. *Int. J. Financ. Stud.* **2020**, *8*, 45. [[CrossRef](#)]
15. Enhholm, I.M.; Papagiannidis, E.; Mikalef, P.; Krogstie, J. Artificial Intelligence and business value: A literature review. *Inf. Syst. Front.* **2022**, *24*, 1709–1734. [[CrossRef](#)]
16. Cob-Parro, A.C.; Losada-Gutiérrez, C.; Marrón-Romera, M.; Gardel-Vicente, A.; Bravo-Muñoz, I. Smart video surveillance system based on edge computing. *Sensors* **2021**, *21*, 2958. [[CrossRef](#)]
17. Rajpurkar, P.; Chen, E.; Banerjee, O.; Topol, E.J. AI in health and medicine. *Nat. Med.* **2022**, *28*, 31–38. [[CrossRef](#)] [[PubMed](#)]
18. Huang, Z.; Shen, Y.; Li, J.; Fey, M.; Brecher, C. A survey on AI-driven digital twins in industry 4.0: Smart manufacturing and advanced robotics. *Sensors* **2021**, *21*, 6340. [[CrossRef](#)]
19. Qazi, S.; Khawaja, B.A.; Farooq, Q.U. IoT-equipped and AI-enabled next generation smart agriculture: A critical review, current challenges and future trends. *IEEE Access* **2022**, *10*, 21219–21235. [[CrossRef](#)]
20. Vincent, D.R.; Deepa, N.; Elavarasan, D.; Srinivasan, K.; Chauhdary, S.H.; Iwendi, C. Sensors driven AI-based agriculture recommendation model for assessing land suitability. *Sensors* **2019**, *19*, 3667. [[CrossRef](#)]
21. Ben Ayed, R.; Hanana, M. Artificial Intelligence to improve the food and agriculture sector. *J. Food Qual.* **2021**, *2021*, 5584754. [[CrossRef](#)]
22. Alla, S.; Adari, S.K.; Alla, S.; Adari, S.K. What is mlops? In *Beginning MLOps with MLFlow: Deploy Models in AWS SageMaker, Google Cloud, and Microsoft Azure*; Apress: New York, NY, USA, 2021; pp. 79–124.
23. John, M.M.; Olsson, H.H.; Bosch, J. Towards mlops: A framework and maturity model. In Proceedings of the 2021 47th Euromicro Conference on Software Engineering and Advanced Applications (SEAA), Palermo, Italy, 1–3 September 2021; pp. 1–8.
24. Ruf, P.; Madan, M.; Reich, C.; Ould-Abdeslam, D. Demystifying mlops and presenting a recipe for the selection of open-source tools. *Appl. Sci.* **2021**, *11*, 8861. [[CrossRef](#)]
25. Hunkeler, U.; Truong, H.L.; Stanford-Clark, A. MQTT-S—A publish/subscribe protocol for Wireless Sensor Networks. In Proceedings of the 2008 3rd International Conference on Communication Systems Software and Middleware and Workshops (COMSWARE'08), Bangalore, India, 6–10 January 2008; pp. 791–798.
26. Berners-Lee, T.; Fielding, R.; Frystyk, H. Hypertext Transfer Protocol—HTTP/1.0. 1996. Available online: <https://www.rfc-editor.org/rfc/rfc1945.html> (accessed on 12 December 2023).
27. Shankar, R.H.; Veeraraghavan, A.; Sivaraman, K.; Ramachandran, S.S. Application of UAV for pest, weeds and disease detection using open computer vision. In Proceedings of the 2018 International Conference on Smart Systems and Inventive Technology (ICSSIT), Tirunelveli, India, 13–14 December 2018; pp. 287–292.
28. Nalluri, S.; Ramasubbareddy, S.; Kannayaram, G. Weather prediction using clustering strategies in machine learning. *J. Comput. Theor. Nanosci.* **2019**, *16*, 1977–1981. [[CrossRef](#)]
29. Ukhurebor, K.E.; Adetunji, C.O.; Olugbemi, O.T.; Nwankwo, W.; Olayinka, A.S.; Umezuruike, C.; Hefft, D.I. Precision agriculture: Weather forecasting for future farming. In *AI, Edge and IoT-Based Smart Agriculture*; Elsevier: Amsterdam, The Netherlands, 2022; pp. 101–121.
30. Kendler, S.; Aharoni, R.; Young, S.; Sela, H.; Kis-Papo, T.; Fahima, T.; Fishbain, B. Detection of crop diseases using enhanced variability imagery data and convolutional neural networks. *Comput. Electron. Agric.* **2022**, *193*, 106732. [[CrossRef](#)]
31. Karunathilake, E.; Le, A.T.; Heo, S.; Chung, Y.S.; Mansoor, S. The path to smart farming: Innovations and opportunities in precision agriculture. *Agriculture* **2023**, *13*, 1593. [[CrossRef](#)]
32. Suryawanshi, Y.; Patil, K. Advancing agriculture through image-based datasets in plant science: A review. *EPRA Int. J. Multidiscip. Res. (IJMR)* **2023**, *9*, 233–236.
33. Agarwal, S.; Hiran, D.; Kothari, H.; Singh, S. Leveraging data processing for optimizing organic farming practices. *J. Data Acquis. Process.* **2023**, *38*, 3243.
34. Rogelja, T.; Ludvig, A.; Weiss, G.; Prah, J.; Shannon, M.; Secco, L. Analyzing social innovation as a process in rural areas: Key dimensions and success factors for the revival of the traditional charcoal burning in Slovenia. *J. Rural. Stud.* **2023**, *97*, 517–533. [[CrossRef](#)]
35. Khudyakova, E.; Shitikova, A.; Stepansevich, M.N.; Grecheneva, A. Requirements of Modern Russian Agricultural Production for Digital Competencies of an Agricultural Specialist. *Educ. Sci.* **2023**, *13*, 203. [[CrossRef](#)]
36. Garske, B.; Bau, A.; Ekardt, F. Digitalization and AI in European agriculture: A strategy for achieving climate and biodiversity targets? *Sustainability* **2021**, *13*, 4652. [[CrossRef](#)]
37. Nova, K. AI-enabled water management systems: An analysis of system components and interdependencies for water conservation. *Eig. Rev. Sci. Technol.* **2023**, *7*, 105–124.

38. Liang, P.; Tang, Y.; Zhang, X.; Bai, Y.; Su, T.; Lai, Z.; Qiao, L.; Li, D. A Survey on Auto-Parallelism of Large-Scale Deep Learning Training. *IEEE Trans. Parallel Distrib. Syst.* **2023**, *34*, 2377–2390. [[CrossRef](#)]
39. Chhedha, S.; Curtis, A.; Siegmann, E.; Chapman, B. Performance Study on CPU-based Machine Learning with PyTorch. In Proceedings of the HPC Asia 2023 Workshops, Singapore, 27 February–2 March 2023; pp. 24–34.
40. Kreuzberger, D.; Kühl, N.; Hirschl, S. Machine learning operations (mlops): Overview, definition, and architecture. *IEEE Access* **2023**, *11*, 31866–31879. [[CrossRef](#)]
41. Hassani, H.; Huang, X.; MacFeely, S. Enabling Digital Twins to Support the UN SDGs. *Big Data Cogn. Comput.* **2022**, *6*, 115. [[CrossRef](#)]
42. Dharmaraj, V.; Vijayanand, C. Artificial Intelligence (AI) in agriculture. *Int. J. Curr. Microbiol. Appl. Sci.* **2018**, *7*, 2122–2128. [[CrossRef](#)]
43. Pistor, N. Accelerating University-Industry Collaborations with MLOps: A Case Study about the Cooperation of Aimo and the Linnaeus University. Master's Thesis, Linnaeus University, Växjö, Sweden, 2023.
44. Yang, X.; Cao, D.; Chen, J.; Xiao, Z.; Daowd, A. AI and IoT-based collaborative business ecosystem: A case in Chinese fish farming industry. *Int. J. Technol. Manag.* **2020**, *82*, 151–171. [[CrossRef](#)]
45. Lubua Dr, E.W. The influence of socioeconomic factors to the use of mobile phones in the agricultural sector of Tanzania. *Afr. J. Inf. Syst.* **2019**, *11*, 2.
46. Kumara, I.; Arts, R.; Di Nucci, D.; Van Den Heuvel, W.J.; Tamburri, D.A. Requirements and Reference Architecture for MLOps: Insights from Industry. *TechRxiv* **2023**. [[CrossRef](#)]
47. Hewage, N.; Meedeniya, D. Machine Learning Operations: A Survey on MLOps Tool Support. *arXiv* **2022**, arXiv:2202.10169.
48. Fujii, T.Y.; Hayashi, V.T.; Arakaki, R.; Ruggiero, W.V.; Bulla, R., Jr.; Hayashi, F.H.; Khalil, K.A. A digital twin architecture model applied with MLOps techniques to improve short-term energy consumption prediction. *Machines* **2021**, *10*, 23. [[CrossRef](#)]
49. Zhou, Y.; Yu, Y.; Ding, B. Towards mlops: A case study of ml pipeline platform. In Proceedings of the 2020 International Conference on Artificial Intelligence and Computer Engineering (ICAICE), Beijing, China, 23–25 October 2020; pp. 494–500.
50. Klerkx, L.; Jakku, E.; Labarthe, P. A review of social science on digital agriculture, smart farming and agriculture 4.0: New contributions and a future research agenda. *NJAS-Wagening. J. Life Sci.* **2019**, *90*, 100315. [[CrossRef](#)]
51. Rose, D.C.; Chilvers, J. Agriculture 4.0: Broadening responsible innovation in an era of smart farming. *Front. Sustain. Food Syst.* **2018**, *2*, 87. [[CrossRef](#)]
52. Placidi, P.; Gasperini, L.; Grassi, A.; Cecconi, M.; Scorzoni, A. Characterization of low-cost capacitive soil moisture sensors for IoT networks. *Sensors* **2020**, *20*, 3585. [[CrossRef](#)]
53. Farooq, M.S.; Riaz, S.; Abid, A.; Abid, K.; Naeem, M.A. A Survey on the Role of IoT in Agriculture for the Implementation of Smart Farming. *IEEE Access* **2019**, *7*, 156237–156271. [[CrossRef](#)]
54. Morar, C.; Doroftei, I.; Doroftei, I.; Hagan, M. Robotic applications on agricultural industry. A review. In *IOP Conference Series: Materials Science and Engineering*; IOP Publishing: Bristol, UK, 2020; Volume 997, p. 012081.
55. Kamilaris, A.; Kartakoullis, A.; Prenafeta-Boldú, F.X. A review on the practice of big data analysis in agriculture. *Comput. Electron. Agric.* **2017**, *143*, 23–37. [[CrossRef](#)]
56. Akkem, Y.; Biswas, S.K.; Varanasi, A. Smart farming using Artificial Intelligence: A review. *Eng. Appl. Artif. Intell.* **2023**, *120*, 105899. [[CrossRef](#)]
57. Siregar, R.R.A.; Seminar, K.B.; Wahjuni, S.; Santosa, E. Vertical farming perspectives in support of precision agriculture using Artificial Intelligence: A review. *Computers* **2022**, *11*, 135. [[CrossRef](#)]
58. Araújo, S.O.; Peres, R.S.; Barata, J.; Lidon, F.; Ramalho, J.C. Characterising the agriculture 4.0 landscape—Emerging trends, challenges and opportunities. *Agronomy* **2021**, *11*, 667. [[CrossRef](#)]
59. Rejeb, A.; Abdollahi, A.; Rejeb, K.; Treiblmaier, H. Drones in agriculture: A review and bibliometric analysis. *Comput. Electron. Agric.* **2022**, *198*, 107017. [[CrossRef](#)]
60. Palei, H. Artificial Intelligence in precision agriculture: A review. *Arch. Comput. Methods Eng.* **2021**, *28*, 1627–1647.
61. Samui, P. Machine Learning Applications in Agriculture: A Comprehensive Review. *Agriculture* **2022**, *12*, 56.
62. Bargoti, S.; Underwood, J. Deep Fruit Detection in Orchards. *arXiv* **2017**, arXiv:1703.08236.
63. Ma, L.; Liu, Y.; Zhang, X.; Ye, Y.; Yin, G.; Johnson, B.A. Deep Learning in remote sensing applications: A meta-analysis and review. *ISPRS J. Photogramm. Remote Sens.* **2019**, *152*, 166–177. [[CrossRef](#)]
64. Prema, P.; Veeramani, A.; Sivakumar, T. Machine Learning Applications in Agriculture. *J. Agric. Res. Technol.* **2022**, *1*, 126. [[CrossRef](#)]
65. Sozzi, M.; Kayad, A.; Ferrari, G.; Zanchin, A.; Grigolato, S.; Marinello, F. Connectivity in rural areas: A case study on internet connection in the Italian agricultural areas. In Proceedings of the 2021 IEEE International Workshop on Metrology for Agriculture and Forestry (MetroAgriFor), Trento-Bolzano, Italy, 3–5 November 2021; pp. 466–470.
66. Visconti, P.; de Fazio, R.; Velázquez, R.; Del-Valle-Soto, C.; Giannoccaro, N.I. Development of sensors-based agri-food traceability system remotely managed by a software platform for optimized farm management. *Sensors* **2020**, *20*, 3632. [[CrossRef](#)]
67. Lee, M.; Wang, Y.R.; Huang, C.F. Design and development of a friendly user interface for building construction traceability system. *Microsyst. Technol.* **2021**, *27*, 1773–1785. [[CrossRef](#)]
68. Dhehibi, B.; Rudiger, U.; Moyo, H.P.; Dhraief, M.Z. Agricultural technology transfer preferences of smallholder farmers in Tunisia's arid regions. *Sustainability* **2020**, *12*, 421. [[CrossRef](#)]

69. Desai, R.; Nisha, T. Best practices for ensuring security in devops: A case study approach. In *Journal of Physics: Conference Series*; IOP Publishing: Bristol, UK, 2021; Volume 1964, p. 042045.
70. Official Jupyter Web Site. Available online: <https://jupyter.org/> (accessed on 19 June 2023).
71. Official Great Expectations Tool Web Site. Available online: <https://greatexpectations.io/> (accessed on 19 June 2023).
72. Official DVC Web Site. Available online: <https://discuss.dvc.org/> (accessed on 19 June 2023).
73. Official MIFlow Web Site. Available online: <https://mlflow.org/> (accessed on 19 June 2023).
74. Smith, J.; Doe, J. Challenges and Opportunities in Internet of Things (IoT): A Comprehensive Survey. *J. IOT Res.* **2021**, *4*, 25–60.
75. Official FlexiGroBot Repository. Available online: <https://github.com/FlexiGroBots-H2020/AI-platform> (accessed on 19 June 2023).
76. FlexiGroBots Entrypoint to Test Kserve Inference Model. Available online: <https://web.platform.flexigrobbots-h2020.eu/apps> (accessed on 19 June 2023).
77. Official FlexiGroBot Kubeflow Access. Available online: <https://kubeflow.flexigrobbots-h2020.eu/> (accessed on 19 June 2023).
78. Official Kubernetes Web Site. Available online: <https://kubernetes.io/> (accessed on 19 June 2023).
79. Official Kubeflow Web Site. Available online: <https://www.kubeflow.org/> (accessed on 19 June 2023).
80. Official Istio Web Site. Available online: <https://istio.io/> (accessed on 19 June 2023).
81. Ladisa, P.; Plate, H.; Martinez, M.; Barais, O. Sok: Taxonomy of attacks on open-source software supply chains. In Proceedings of the 2023 IEEE Symposium on Security and Privacy (SP), San Francisco, CA, USA, 21–25 May 2023; pp. 1509–1526.
82. Official Kubeflow Pipeline Web Site. Available online: <https://www.kubeflow.org/docs/components/pipelines/v1/introduction/> (accessed on 19 June 2023).
83. Official TensorFlow Web Site. Available online: <https://www.tensorflow.org/> (accessed on 19 June 2023).
84. Official Pytorch Web Site. Available online: <https://pytorch.org/> (accessed on 19 June 2023).
85. Official Sklearn Web Site. Available online: <https://scikit-learn.org/stable/> (accessed on 19 June 2023).
86. Official Docker Web Site. Available online: <https://www.docker.com/> (accessed on 19 June 2023).
87. Official Katib Web Site. Available online: <https://www.kubeflow.org/docs/components/katib/overview/> (accessed on 19 June 2023).
88. Official TFX Web Site. Available online: <https://www.tensorflow.org/tfx/guide/serving> (accessed on 19 June 2023).
89. Official Seldon Web Site. Available online: <https://www.seldon.io/> (accessed on 19 June 2023).
90. Official Kserve Web Page. Available online: <https://www.kubeflow.org/docs/external-add-ons/kserve/kserve/> (accessed on 17 August 2023).
91. Choi, K.; Yi, J.; Park, C.; Yoon, S. Deep Learning for anomaly detection in time-series data: Review, analysis, and guidelines. *IEEE Access* **2021**, *9*, 120043–120065. [\[CrossRef\]](#)
92. Official Camel Web Site. Available online: <https://camel.apache.org/> (accessed on 19 June 2023).
93. Jahanshahi, H.; Alijani, Z.; Mihalache, S.F. Towards Sustainable Transportation: A Review of Fuzzy Decision Systems and Supply Chain Serviceability. *Mathematics* **2023**, *11*, 1934. [\[CrossRef\]](#)
94. Tomasiello, S.; Alijani, Z. Fuzzy-based approaches for agri-food supply chains: A mini-review. *Soft Comput.* **2021**, *25*, 7479–7492. [\[CrossRef\]](#)
95. Official Grafana Web Site. Available online: <https://grafana.com/> (accessed on 19 June 2023).
96. Official Prometheus Web Site. Available online: <https://prometheus.io> (accessed on 19 June 2023).

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.