

DigiAgriApp: a client-server application to monitor field activities

Marco Moretto^{a,*}, Luca Delucchi^a, Roberto Zorer^a, Damiano Moser^b, Franco Micheli^b, Andrea Paoli^c, Pietro Franceschi^a

^a Unit of Digital Agriculture, Research and Innovation Centre, Fondazione Edmund Mach, Via E. Mach 1, San Michele all'Adige, 38098, Italy

^b Unit of Fruit Experimental Cultivation, Technological Transfer Centre, Fondazione Edmund Mach, Via E. Mach 1, San Michele all'Adige, 38098, Italy

^c Wolfson School, Loughborough University, Wolfson Building, Ashby Rd, Leicestershire, Loughborough, LE11 3TU, United Kingdom

ARTICLE INFO

Keywords:

Digital agriculture
Precision farming
Remote sensing
Database
REST API
Python

ABSTRACT

Farming is increasingly data-driven, leveraging high-frequency and precision data from IoT devices, sensors, and remote tools. Effective data collection, organization, and management are essential to link datasets with agronomic details, forming the foundation for predictive models. These models, using AI and machine learning, optimize decision-making, forecast crop yields, predict pest outbreaks, and enhance resource use. High-quality, diverse data integration is key to building accurate tools that address agriculture's complexity, boosting productivity and resilience. We introduce DigiAgriApp, an open-source client-server application for centralized farming data management. It tracks crop details, sensor readings, irrigation, field operations, production statistics, and emissions for Life Cycle Assessment. Initially developed for the Fondazione Edmund Mach, DigiAgriApp has evolved into a versatile tool. Users can access a public server or deploy a private instance via Docker, making it ideal for institutions, farmers, and corporations alike.

DigiAgriApp is available at <https://digiagriapp.gitlab.io/digiagriapp-website/>.

1. Introduction

Agriculture is recognized as a critical component of the economic and social infrastructure, playing a vital role in ensuring food production and security. As the world's primary food source (Ritchie et al., 1970), agriculture has been instrumental in the evolution of civilizations throughout history. Presently, we are witnessing the advent of 'Agriculture 4.0', a term coined to encapsulate the exploration and application of Information and Communication Technology (ICT) and advanced data analytics (e.g. Artificial Intelligence) to enhance agricultural processes (Liu et al., 2021).

Digital Agriculture (DA), characterized by the application of data-centric strategies to enhance agricultural productivity and sustainability, is perceived as a solution for improving farm efficiency in food production and, for this reason, the field of data-oriented agriculture is witnessing an expansion in research (Basso and Antle, 2020). Indeed, there has been a growing interest in recent years towards the tools and applications associated with DA (Chin et al., 2024; Oliveira et al., 2014; Oteyo et al., 2021; Kuntke et al., 2023; MacPherson et al., 2022; Friha et al., 2021).

Within the realm of Big Data (De Mauro et al., 2016), which refers to

the extensive quantities of diverse data that can be collected, examined, and utilized for decision-making in agriculture, Wolfert (Wolfert et al., 2017) and colleagues identified six "state of the art of big data applications in smart farming", namely: data capture, storage, transfer, transformation, analytics and marketing. Data, in its broadest sense, serves as the backbone of informed decision-making, and in the realm of agriculture, meticulous data collection is pivotal in driving sustainable practices, optimizing yields, and ensuring food security on a global scale.

Data organization and interoperability are also the prerequisite for the large-scale application of powerful and data-eager models (e.g. AI), which can be used to link agronomic practice and "big data". In this context, the application of the "open data" philosophy is a cornerstone and it should be coupled with open-source infrastructure to ensure long-term accessibility and interoperability. Such systems must prioritize user-friendliness, catering to the day-to-day needs of farm workers who interact with these platforms regularly. The combination of robust data management, open standards, and intuitive interfaces creates a framework that not only facilitates advanced predictive modeling but also empowers agricultural practitioners to make data-driven decisions efficiently.

* Corresponding author.

E-mail address: marco.moretto@fmach.it (M. Moretto).

In this paper we describe the implementation of a tool, named DigiAgriApp, used to centralize farming data management. It permits managing different farms and for each of them it is possible to add fields, sub-fields, row and plant. The application enables users to manage various crops, operations, treatments, observations, measurements, as well as data gathered from sensors, proximal and/or remote sensing. All this information can be associated with the fields or other components such as sub-fields, rows, and plants.

2. Methods

2.1. Software architecture

DigiAgriApp is a client-server application, with the server component comprising a relational database enhanced with a spatial extension. This extension adds functionalities to the database, such as support for storing, indexing, and querying geospatial data, enabling more effective management of location-based information. Moreover, the server features a REST API for seamless communication between clients and the database (see Fig. 1).

The database is designed for easy adaptation to various situations, projects, and users, catering to agricultural settings of all sizes, from home gardens to large farms. It consists of more than 50 tables, with the 'Farm' and 'Field' tables serving as central hubs due to their high number of connections within the database. The "Farm" table is utilized to grant users access to their specific objects, such as sensors, devices, and fields. The "Field" table, on the other hand, serves as a hub connecting all other information, starting from field elements like sub-fields, rows, and plants, and their associated data, which could be sensor readings, remote sensing, production data or agronomic descriptors. Information about cultivars, irrigation, and production can be assigned not only to the field object but also more granularly to sub-fields, rows, or individual plants.

To ensure coherence among agronomic descriptors, DigiAgriApp limits the use of free text input. Instead, users can define new terminology hierarchies to be used within the application, allowing for standardization while maintaining flexibility for specific needs. This approach helps maintain data consistency and facilitates more accurate analysis and reporting. Moreover, DigiAgriApp comes preloaded with default standard terminology, which is used to populate database tables

after the initial application installation. This default terminology provides a solid foundation for users, ensuring a common language for agricultural data across different implementations while still allowing for customization.

The chosen object-relational DataBase Management System (DBMS) is PostgreSQL (<https://www.postgresql.org/>), with the geographical extension PostGIS (<https://postgis.net/>) employed within PostgreSQL to support GIS data and functions. Vector format is used to store geometries of fields, rows, and plants, while raster format is used for remote sensing data. The server is developed using Django (<https://www.djangoproject.com/>), a high-level Python web framework. The Django REST framework extension, along with other minor extensions, is used to create the REST API. This service acts as a crucial interface between the database and the client. We have opted for a nested structure for the API, with the farm as the primary element. This allows the user to view only the farms related to them and from there, access other nested elements, starting with the farm's fields and other elements like sensor, remote data as well as other subfields like rows and plants. The REST API uses JavaScript Object Notation (JSON) as the input and output format to simplify and standardize communication.

In order to retrieve data from the sensors, the server is equipped with an expanding suite of services to collaborate with data providers, though only a few are currently implemented.

The Message Queue Telemetry Transport (MQTT) provider operates as a daemon, perpetually listening to a broker (a backend system that coordinates various clients) and multiple topics to acquire data as soon as it becomes available. MQTT is a lightweight protocol, specifically designed for connections involving devices with resource constraints or limited network bandwidth, making it a frequently used protocol in IoT solutions. We leverage The Things Stack LoRaWAN network as a service to connect our sensors using LoRaWAN and provide the data through the MQTT service offered by The Things Stack. Data derived from remote sensing utilizes the SpatioTemporal Asset Catalogs (STAC) specification. STAC serves as a universal language to describe geospatial information, facilitating its manipulation, indexing, and discovery. The STAC provider consists of a general class, atop which a provider is created using the Earth Search service, a collaboration between Element84 (<https://www.element84.com/>) and Amazon AWS (<https://aws.amazon.com/>). By utilizing the collection with the COG (Cloud Optimized GeoTIFF) format, a specialized GeoTIFF format optimized for internet access, it is

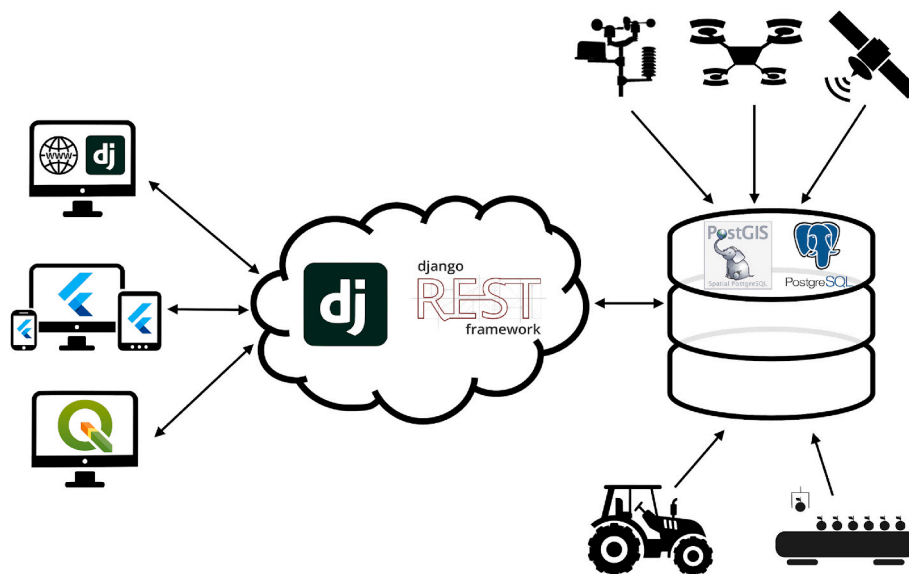


Fig. 1. The overall structure of DigiAgriApp, consisting of the Django backend and API layer. DigiAgriApp is designed to collect data from multiple sources, including IoT devices, machinery, and remote sensing; store and process this data using a powerful, geo-enabled database; provide access to the data and functionality through various interfaces, including web, mobile, and desktop applications; and integrate with existing GIS software for advanced spatial analysis.

possible to download only the necessary pixels of a field, thereby enhancing performance and expediting operations on that data.

Most tables offer a simple method to import and export data as CSV files. This can be done by creating a CSV file that matches the table structure or using more complex procedures for specific data, like importing individual plant records.

DigiAgriApp comprises two clients: a multiplatform application and a QGIS (QGIS Geographic Information System, 2024) plugin, an extension for the open-source QGIS software installable via the QGIS Plugin Manager (see Fig. 2). The primary client is developed using Flutter (<https://flutter.dev/>), an open-source UI software development kit, which is built on Dart (<https://dart.dev/>), a programming language specifically designed for client development. Flutter's ability to create cross-platform applications for Android, iOS, Linux, macOS, Windows, and the web from a single codebase was the primary reason for its selection.

Comprehensive instructions for installing and using DigiAgriApp are available on the application's website: <https://digiagriapp.gitlab.io/digiagriapp-website/>. The site also provides guidance on setting up both the client and server applications for local use, along with details on accessing the public DigiAgriApp instance.

2.2. Software functionalities

The DigiAgriApp client is designed to receive, display, alter, and generate new data, which can subsequently be transmitted back to the server. Upon initiating the application and inserting the required credentials, users are greeted with a list of their associated farms. Choosing a farm directs the user to a display of the fields that belong to that farm. Each field is depicted by a static map of the region, supplemented by pertinent information and statistics. When a field is selected, a new screen emerges (see Fig. 3), providing an exhaustive overview of the field. This includes a dynamic map that delineates the field's boundary, its associated sub-fields, rows, and if applicable, plants. An information panel offers active specifics such as the current cultivars or irrigation systems. The bottom section of the page houses a toolbar with multiple buttons, each leading to a new tab that exhibits selected elements of the field. Users have the flexibility to toggle between different datasets encompassing operations, treatments, observations, measurements, among others. Moreover, a dedicated tab presents charts with data derived from sensors or remote sensing.

DigiAgriApp supports the rapid and precise acquisition of biometric

parameters, such as trunk diameter, to enhance the monitoring of plant growth and health. Currently, these measurements are manually recorded by field operators. However, sensors are already capable of transmitting data to the server via the ReST API endpoint. In the future, the integration of direct interfaces with specific sensors may further streamline data acquisition.

Moreover, DigiAgriApp includes a feature called Pest Patrol, a specialized plugin designed for the automatic recognition of two significant pest vectors, *Scaphoideus titanus* and *Orientus ishidae*. This feature is based on a recent study (Checola et al., 2024) that employs computer vision techniques. Pest Patrol utilizes advanced object detection algorithms to accurately identify these pests from images captured directly with a mobile phone.

Additionally, users can leverage GIS tools within DigiAgriApp to correlate production data with soil characteristics, offering insights into the impact of soil on crop performance. The app holds potential for forecasting crop diseases by analyzing historical and real-time data, helping to anticipate and mitigate risks.

User access is differentiated by roles: normal users are limited to managing their own farms, while admin users can add or remove permissions for normal users, enabling broader management capabilities. The admin panel, powered by Django, allows admins to manage user access and permissions. Note that an internet connection is required for the application to operate, as it synchronizes with the server to retrieve and update data continuously.

2.3. Use case

The initial deployment of the DigiAgriApp platform occurred within the Fondazione Edmund Mach. Its first use was by the Fruit Cultivation Experimental Unit for transitioning from an obsolete legacy data storage system to DigiAgriApp. The Unit of Fruit Experimental Cultivation manages research activities, primarily focused on apple trees, across three farms spanning approximately 15 ha and currently comprising over 20,000 apple trees. About 10 percent of these trees are kept under observation, with various measurements taken and, most importantly, all harvested apples sorted to analyze production data of plants under different experimental conditions. The primary focus of the trials includes the assessment of yield performance across various training systems, planting densities, the evaluation of the effects of different horticultural management practices, such as pruning and thinning, and the analysis of the impact of diverse rootstocks on apple tree

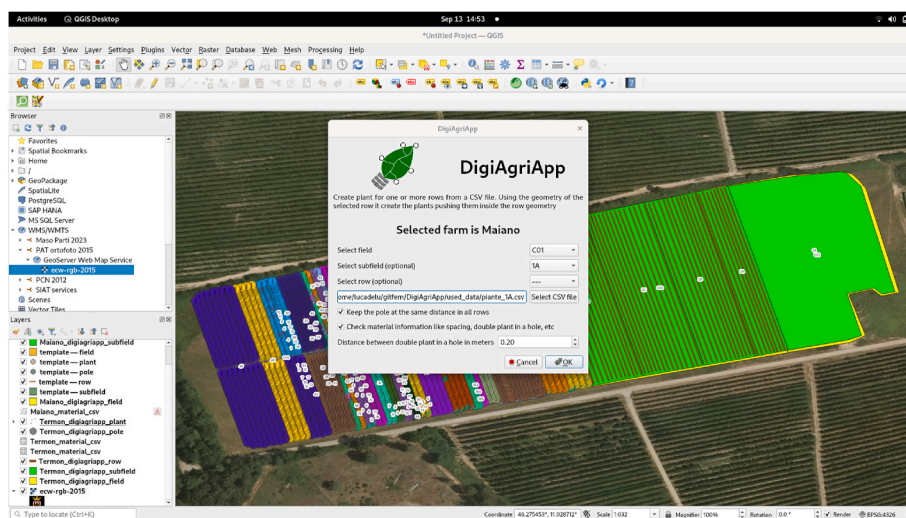


Fig. 2. One of the options in the DigiAgriApp QGIS plugin allows for creating plant locations for one or more rows from a CSV file. The fields are color-coded to represent different management zones. The plugin also offers options to maintain consistent pole distances and set the distance between double plants in a single hole. (For interpretation of the references to color in this figure legend, the reader is referred to the Web version of this article.)

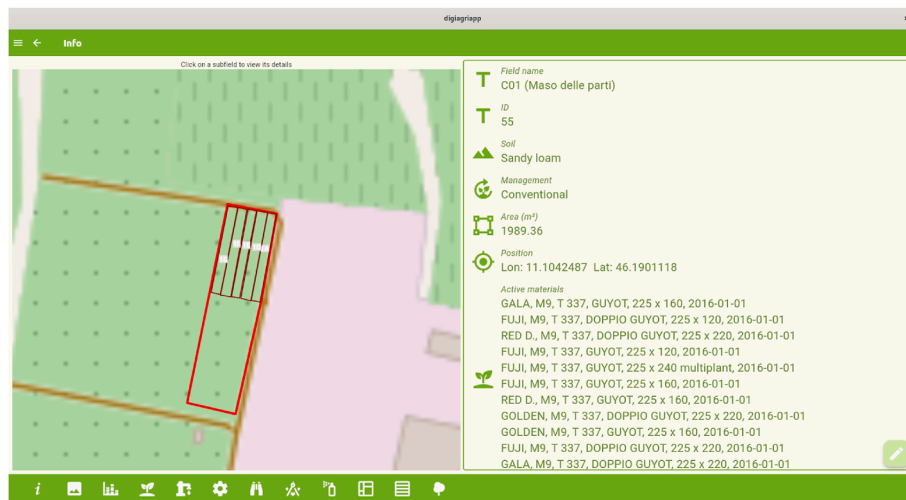


Fig. 3. A detailed view of one of the DigiAgriApp GUI panels. On the left side is the map view, displaying agricultural fields with subfields and rows highlighted in red. On the right is the information panel, providing detailed data such as the field name, ID, soil type, management type, area, and geographic position. It also shows active material information, including varieties, rootstocks, training system, trellis type, planting dimensions, and planting date. At the bottom is the toolbar, featuring icons for various application functions (e.g., charts, plant information, settings, etc.). (For interpretation of the references to color in this figure legend, the reader is referred to the Web version of this article.)

productivity. Currently we have recorded a total of 62 rootstocks, 33 apple varieties, and 132 planting distances tested in the DigiAgriApp database. The experimental orchard registry currently covers an area of 5.5 ha. It contains a total of 19,664 trees along with their associated materials. However, the plants and their associated materials can vary significantly within a single plot or section of the orchard. Additionally, we are tracking 3042 apple trees identified through labeling. For these, we are focusing on the productive parameters obtained from a Greefa MSE 2000 digital sorter and autumn measurements of trunk circumferences. During the migration process, new functionalities were incorporated into the application. First, the database was extended to save production data, which was the most lacking aspect. Subsequently, functions were developed to load data from individual apples and then to export data grouped by plant, year, and experimental conditions. The transition began with a comprehensive mapping of the farms into the DigiAgriApp database, georeferencing fields, sub-fields, and rows using the QGIS plugin. Information on the material present in different fields was then added to the database, including variety, clone, rootstock, planting date, training system, and distances between plants and rows. The next step was to georeference all plants starting from a CSV file, using the dedicated functionality of the QGIS plugin, which can distribute plants along associated rows. Following this, diameter or circumference values of the trunk for plants under observation were entered using CSV files compiled with templates and uploaded via the web administration panel. All this information is used to relate production data from the grader to the characteristics of a tree, e.g., its production based on trunk circumference or potential production per hectare for a given material.

During the migration from the previous data storage system to DigiAgriApp, suggestions from colleagues played a key role in shaping the development and enhancement of both the client and the administration panel. Over the past year, several new features have been implemented to improve the user experience. These include simplified form filling for adding or modifying data, the display of attributes linked to various elements such as fields, plots, rows, and plants, quick actions for managing multiple trees (such as observations, measurements, and material changes), the ability to display specific plant information on the map, and the export of data from the admin panel in CSV format. Additionally, several new functionalities are planned for the coming months, including the introduction of caching mechanisms for both the server and the application to improve performance, new widgets for form

filling (such as a tap widget for counting elements on the mobile screen and drag-and-drop functionality for managing plant data), and enhanced display options for plants based on crop-related information. User feedback remains a central element of our development process, with new features continuously added in response to the evolving needs of users. Furthermore, as DigiAgriApp is open-source and hosted on GitLab, we plan to leverage the platform's feedback capabilities, including issues and feature requests, to allow users to directly contribute their suggestions, report issues, and propose new features to guide future developments of DigiAgriApp.

The migration to the new application offered several advantages, including.

- Comprehensive plant status tracking, such as disease identification and prevalence
- Efficient management of plant mortality records
- Automated report generation, including data aggregation by tree or experiment and visualization through graphical representations

2.4. Results and discussion

DigiAgriApp is a client-server application that is free and open-source, aimed at centralizing and optimizing data management in agriculture. It allows users to organize and analyze various datasets, including crop information, sensor data, irrigation schedules, field operations, remote sensing imagery, production statistics, and emissions data. The initial development of DigiAgriApp was driven by a specific institutional need, namely, the management of research activities primarily focused on apple trees. However, flexibility was a key principle in its design, ensuring that its functionality extends beyond tree-based systems. DigiAgriApp can be adapted to various crop types, enabling the management of fine-grained entities ranging from individual plant clones to entire fields. Users can define the scale at which activities and measurements are monitored. For instance, row crops such as maize, wheat, and soybeans can be managed at the sub-field or field level. Although data analysis is conducted on DigiAgriApp-generated data, its primary focus is on applied purposes rather than research objectives. DigiAgriApp is designed primarily for commercial farmers, while also accommodating researchers. To maintain an efficient and lightweight data model, we deliberately chose not to implement phenotyping-specific features, such as BrAPI (Breeding API, <https://brapi.org/>)

compliance or adherence to the MIAPPE standard (<https://www.miappe.org/>), although we implemented all the mandatory fields. Instead, the platform prioritizes functionalities that align with the practical needs of farmers.

The application features a user-friendly interface, accessible through a cross-platform Flutter application and a QGIS plugin, which provides tools for data visualization, management, and analysis at detailed levels, including fields, subfields, rows, and individual plants. DigiAgriApp is designed to support large-scale, multi-farm implementations, providing a centralized platform for efficient management of multiple agricultural operations. The backend, constructed with Django and PostgreSQL, ensures scalability and robustness for handling substantial datasets. The architecture is designed to adapt to the growth of agricultural operations, maintaining performance and data integrity across diverse farm sizes. However, as is characteristic of scalable systems, performance and responsiveness are subject to the limitations of the deployment environment, including server resources, database optimization, and network connectivity. DigiAgriApp supports integration with IoT devices and remote sensing platforms, facilitates data import/export, and uses standardized terminology to maintain consistency. Its extendable architecture and emphasis on user accessibility make it suitable for a wide range of agricultural practices, enhancing decision-making and productivity. The DigiAgriApp multi-platform application serves as the primary interface for the DigiAgriApp server. Initial setup requires either new server registration or existing account sign-in. Upon successful connection, users can select from available farms. The main interface displays associated fields, viewable as a list or map, along with linked machinery and experiments. Selecting a field provides access to its detailed information, including map, plots, rows, area, plant count,

and active materials. A toolbar facilitates navigation between screens, displaying field-specific information. Users can select and view details for plots, rows, and individual plants. The main menu offers quick access to field activities such as reporting plant mortality, adding plant labels, and recording observations or measurements. Configuration requires server details, user credentials, and farm selection. The plugin enables.

1. Exporting geographical elements from the DigiAgriApp database
2. Importing vector and raster data from QGIS layers into DigiAgriApp
3. Creating plant maps from CSV files, maintaining equidistant spacing based on plant and pole information
4. Styling data according material information

In the context of agricultural management software, several platforms offer functionalities comparable to DigiAgriApp, each with distinct characteristics and target demographics (Basir et al., 2024). This comparative analysis focuses on FarmOS (farmOS: A web, 2024), Tania (Tania: a farm management software, 2024), LiteFarm (LiteFarm: a free and open, 2024), and xFarm (<https://xfarm.ag/>), with Table 1 summarizing their key features and differences. FarmOS, built on the Drupal framework, provides extensive customization and scalability, making it suitable for large-scale agricultural operations. However, its implementation requires significant technical expertise. The platform's flexibility allows for integration with various agricultural technologies and data sources, facilitating comprehensive farm management. Notably, FarmOS offers a limited mobile experience, as it is primarily designed as a web application, potentially constraining its utility in field operations. Tania, in contrast, offers a lightweight and user-friendly interface, optimized for small-scale farms or home gardens. While its simplicity

Table 1
Comparative analysis of DigiAgriApp, FarmOS, Tania, LiteFarm, and xFarm based on various features.

Feature	DigiAgriApp	FarmOS	Tania	LiteFarm	xFarm
Target User/Farm Size	All sizes (home to large), institutions	Large-scale agricultural operations	Small-scale farms, home gardens	Small to medium-sized sustainable farms	Varying sizes
Key Features/Strengths	Centralized data, open-source, customizable, multiplatform (web, mobile, QGIS), data integration	Extensive customization, scalability, integration with various technologies	Lightweight, user-friendly	Balance of usability and functionality, mobile-first design, sustainability focus	Versatile, accessible, robust data integration, sustainability features
Limitations/Weaknesses	Internet required for continuous sync (full offline capabilities planned)	Requires significant technical expertise, limited mobile experience	Lacks advanced features and scalability, limited mobile functionality	Less customization than FarmOS	Advanced tools in paid tiers, free tier limited
Mobile Accessibility	Yes	Limited (web-based)	Limited (web-based)	Yes	Yes
Pricing	Free	Free	Free	Free	Tiered pricing
Open-Source	Yes	Yes	Yes	Yes	No
Actively developed	Yes	Yes	No code updates in 3 years	Yes	Yes
Working offline	No	No	No	No	No
Support for georeferenced fields, subfields, rows and plants	Yes	Only fields and plants	No	Only fields	Only fields
Observations	Yes	Limited (Note)	No	Limited (Task)	Limited (Note)
Measurements	Yes	Yes	No	Limited (Task)	Limited (Note)
Treatments	Yes	Limited (Task)	Limited (Note)	Yes	Yes
Operations	Yes	Limited (Note)	Limited (Task)	Yes	Yes
Irrigation	Yes	No	Only water capacity	Yes	Only paid tiers
Sensors data	Yes	No	No	No	Yes
Experiments	Yes	No	No	No	No
Harvest	Yes	Yes	No	Yes	Yes
Remote Sensing data	Yes	No	No	No	Yes
Sorting machine data	Yes	No	No	No	No
Managing finances	No	No	No	Yes	Only paid tiers
Storage room	No	No	No	Yes	Yes
Multi user	Yes	Yes	No	Yes	Yes
Animals	No	Yes	No	Yes(in beta)	Yes
Certification	No	No	No	Yes	Yes
Insect detection	Yes (2 species)	No	No	No	Only paid tiers (20 species)
Machines	Yes	Yes	No	No	Yes
Tasks	No	No	Yes	Yes	Yes

enhances accessibility, it lacks advanced features and scalability found in more robust systems. This limitation restricts its applicability in complex or expanding agricultural operations. Similar to FarmOS, Tania's mobile functionality is restricted due to its web-based architecture, which may impede its effectiveness for on-site data entry and management. LiteFarm targets small to medium-sized sustainable farms, striking a balance between usability and functionality. Its mobile-first design enhances field accessibility, though it offers less customization compared to FarmOS. The platform's focus on sustainability includes features for environmental impact assessment and biodiversity management. xFarm offers versatility and accessibility, along with robust data integration and sustainability features, catering to farms of varying sizes. However, its tiered pricing structure restricts advanced tools to paid subscribers, leaving the free tier limited to basic functionalities. These are not the only available options; other software solutions with similar capabilities exist, but they are typically offered under commercial licenses and are designed to manage a single crop. Examples include EnoGIS (<https://www.enogis.it/>) and 4Grapes (<https://www.academy.4grapes.it/>), both of which are tailored specifically for grapevine management.

DigiAgriApp, on the other hand, distinguishes itself by combining ease of installation by using Docker to enable seamless and efficient setup across different environments, with a comprehensive feature set, while also being designed for expandability, allowing users to add new functionalities as their needs grow.

In terms of data security and privacy, DigiAgriApp adheres to General Data Protection Regulation (GDPR) standards concerning data security and privacy. It achieves this by implementing user consent mechanisms and ensuring the rights to data portability and erasure. Data transmitted between users and the application is secured using Transport Layer Security (TLS) encryption, thereby preventing unauthorized access. The application utilizes Django's built-in authentication mechanisms, including hashed password storage (PBKDF2) and role-based access control (RBAC), ensuring users can only access data pertinent to their roles. Additionally, it integrates Django's Object-Relational Mapper (ORM) and templating engine to provide inherent protection against SQL injection and cross-site scripting (XSS) attacks. To counter cross-site request forgery (CSRF) attacks, Django's CSRF middleware is also employed. Moreover, DigiAgriApp supports user data anonymization upon deletion, ensuring the removal of personally identifiable information while preserving anonymized data for research purposes. These security and privacy measures, along with its flexibility and scalability, establish DigiAgriApp as a robust solution for diverse agricultural management needs.

3. Conclusion and future work

DigiAgriApp is an application designed to streamline the process of data management within the agricultural sector. The effectiveness of any decision support tool is measured by its ability to convert recommendations into successful actions, with data collection and management being the initial step. DigiAgriApp fulfills this role by providing a suite of tools that enable various use-case scenarios, including intelligent data transmission and direct server-client data retrieval of crop and sensor data, irrigation details, field operations, remote sensing imagery, and production statistics. Our solution employs a server-client architecture to manage information over the internet, utilizing standard and open-source technologies. We have demonstrated a real-world scenario where DigiAgriApp is employed in production to gather and monitor various field activities. Tools like DigiAgriApp play a crucial role in developing robust and effective prediction models for agriculture. By providing comprehensive and accurate data collection and management capabilities, these tools enable researchers and farmers to build more reliable predictive models, which in turn lead to better-informed decision-making and improved agricultural outcomes. The quality and quantity of data gathered through such platforms significantly enhance

the accuracy and applicability of agricultural forecasting and management strategies. DigiAgriApp is currently maintained and updated regularly. Despite being used in a production environment, it is still a work in progress, continually evolving and expanding its feature set in response to user requests. Looking to the future, the development team is planning to enhance DigiAgriApp's functionalities, one of the most demanding will be to implement full offline capabilities. This upcoming feature will allow users to utilize the application seamlessly even when working in areas with limited or no internet connectivity, ensuring uninterrupted data collection and management in the field. This offline functionality will significantly increase the app's versatility and usefulness, particularly in remote agricultural settings. This feature was not implemented from the outset, as it requires thorough evaluation and testing, while the primary objective was to ensure the application's immediate functionality. To enable offline use, several critical aspects must be carefully addressed, as they may introduce security vulnerabilities or data consistency issues. The development process involves storing a local copy of the data with timestamps and synchronizing it with the backend using a Last-Write-Wins (LWW) policy to resolve potential conflicts. This approach ensures a straightforward implementation and enables rapid conflict resolution with minimal computational overhead. However, this approach may result in unintentional data loss if the last write overrides changes from previous updates. Nevertheless, the risk of data overwrites is expected to be minimal, as multiple farmers are unlikely to modify the same entity within the same field simultaneously.

Additionally, the team plans to integrate and adopt terminology from the Open Agri Common Semantic Model (<https://github.com/openagri-i-eu/OCSM>) and the Agriculture Information Model currently being developed by the Open Geospatial Consortium (<https://github.com/opengeospatial/Agriculture-DWG>). By aligning with the standards, DigiAgriApp will foster compatibility and data interoperability within the broader agricultural technology ecosystem.

CRedit authorship contribution statement

Marco Moretto: Writing – review & editing, Writing – original draft. **Luca Delucchi:** Writing – review & editing, Validation, Software, Conceptualization. **Roberto Zorer:** Writing – review & editing. **Damiano Moser:** Writing – review & editing, Validation, Conceptualization. **Franco Micheli:** Validation, Conceptualization. **Andrea Paoli:** Conceptualization. **Pietro Franceschi:** Writing – review & editing, Supervision.

Software availability

Software Name: DigiAgriApp.
 Developer: Fondazione Edmund Mach.
 Technical support: Luca Delucchi (luca.delucchi@fmach.it)
 Programming language: Python, Flutter.
 License: All code released under DigiAgriApp umbrella is released under the GPL v3.

Software required: Python 3, PostgreSQL, PostGIS, Proj4, GDAL, redis, plus Python Libraries <https://gitlab.com/digiagriapp/digiagriapp-server/-/blob/main/requirements.txt>.

Software website: <https://digiagriapp.gitlab.io/digiagriapp-website/>

Documentation: <https://digiagriapp.gitlab.io/digiagriapp-documentation>.

Presentations: <https://digiagriapp.gitlab.io/digiagriapp-presentations/>

Declaration of generative AI and AI-assisted technologies in the writing process

During the preparation of this work the authors used Microsoft

Copilot to improve grammar and spelling. After using this tool/service, the authors reviewed and edited the content as needed and take full responsibility for the content of the publication.

Funding

This work was supported by the Autonomous Province of Trento ADP P2211007I, the 10° and 12° Calls of the Fondazione VRT, Valorizzazione della Ricerca Trentina, the Interconnected Nord-Est Innovation Ecosystem (iNEST), and the AgrifoodTEF project under the Digital Europe Programme (Grant Agreement No. 101100622; CUP: C63C22001180007).

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

The authors would like to acknowledge the colleague Paolo Sonogo for the constructive feedback which strengthened the clarity and consistency of this research paper.

Abbreviations

AI	Artificial Intelligence
API	Application Programming Interface
AWS	Amazon Web Services
COG	Cloud Optimized GeoTIFF
CSV	Comma-Separated Values
DA	Digital Agriculture
DBMS	DataBase Management System
GIS	Geographic Information System
ICT	Information and Communication Technology
IoT	Internet of Things
JSON	JavaScript Object Notation
MQTT	Message Queue Telemetry Transport

OCSM	Open Agri Common Semantic Model
REST	Representational State Transfer
STAC	SpatioTemporal Asset Catalog

Data availability

No data was used for the research described in the article.

References

- Basir, Md S., Buckmaster, D., Raturi, A., Zhang, Y., 2024. From pen and paper to digital precision: a comprehensive review of on-farm recordkeeping. *Precis. Agric.* 25, 2643–2682.
- Basso, B., Antle, J., 2020. Digital agriculture to design sustainable agricultural systems. *Nat. Sustain.* 3, 254–256.
- Checola, G., et al., 2024. A novel dataset and deep learning object detection benchmark for grapevine pest surveillance. *Front. Plant Sci.* 15.
- Chin, S.-W., Rubambiza, G., Zhao, Y., Malek, K., Weatherspoon, H., 2024. Realtime optimization and management system (ROAM): a decision support system for digital agriculture systems. *Smart Agric. Technol.* 8, 100452.
- De Mauro, A., Greco, M., Grimaldi, M., 2016. A formal definition of Big Data based on its essential features. *Libr. Rev.* 65, 122–135.
- farmOS: a web-based farm record keeping application. <https://farmos.org/>, 2024.
- Friha, O., Ferrag, M.A., Shu, L., Maglaras, L., Wang, X., 2021. Internet of Things for the future of smart agriculture: a comprehensive survey of emerging technologies. *IEEECAA J. Autom. Sin.* 8, 718–752.
- Kuntke, F., Kaufhold, M.-A., Linsner, S., Reuter, C., 2023. GeoBox: design and evaluation of a tool for resilient and decentralised data management in agriculture. *Behav. Inf. Technol.* 1–23. <https://doi.org/10.1080/0144929X.2023.2185747>.
- LiteFarm: a free and open source farm management tool built for current and aspiring sustainable farmers. <https://www.litefarm.org/>, 2024.
- Liu, Y., Ma, X., Shu, L., Hancke, G.P., Abu-Mahfouz, A.M., 2021. From industry 4.0 to agriculture 4.0: current status, enabling technologies, and research challenges. *IEEE Trans. Ind. Inform.* 17, 4322–4334.
- MacPherson, J., et al., 2022. Future agricultural systems and the role of digitalization for achieving sustainability goals. A review. *Agron. Sustain. Dev.* 42, 70.
- Oliveira, T. H. M. de, Painho, M., Santos, V., Sian, O., Barriguinha, A., 2014. Development of an agricultural management information system based on open-source solutions. *Procedia Technol.* 16, 342–354.
- Oteyo, I.N., Marra, M., Kimani, S., Meuter, W.D., Boix, E.G., 2021. A survey on mobile applications for smart agriculture. *SN Comput. Sci.* 2, 293.
- QGIS Geographic Information System, 2024. Open Source Geospatial Foundation Project. <http://qgis.org.QGIS.org>.
- Ritchie, H., Rosado, P., Roser, M., 1970. Agricultural Production. *Our World Data*.
- Tania: a farm management software for the hobbyist and smallholder farmer. <https://usetania.org/>, 2024.
- Wolfert, S., Ge, L., Verdouw, C., Bogaardt, M.-J., 2017. Big data in smart farming – a review. *Agric. Syst.* 153, 69–80.