



FARMIT: continuous assessment of crop quality using machine learning and deep learning techniques for IoT-based smart farming

Ángel Luis Perales Gómez¹ · Pedro E. López-de-Teruel¹ · Alberto Ruiz² · Ginés García-Mateos² · Gregorio Bernabé García¹ · Félix J. García Clemente¹

Received: 4 May 2021 / Revised: 5 September 2021 / Accepted: 24 November 2021 / Published online: 31 March 2022
© The Author(s) 2021

Abstract

The race for automation has reached farms and agricultural fields. Many of these facilities use the Internet of Things technologies to automate processes and increase productivity. Besides, Machine Learning and Deep Learning allow performing continuous decision making based on data analysis. In this work, we fill a gap in the literature and present a novel architecture based on IoT and Machine Learning / Deep Learning technologies for the continuous assessment of agricultural crop quality. This architecture is divided into three layers that work together to gather, process, and analyze data from different sources to evaluate crop quality. In the experiments, the proposed approach based on data aggregation from different sources reaches a lower percentage error than considering only one source. In particular, the percentage error achieved by our approach in the test dataset was 6.59, while the percentage error achieved exclusively using data from sensors was 6.71.

Keywords Crop quality · Deep learning · Internet of things · Machine learning · Smart farming

1 Introduction

Automation has reached all areas of our society, and farms and agriculture are no exception. In this context, more and more farms and agricultural fields contain some type of automation to increase the performance of their production

processes [15]. This increase in automation, together with the arrival of mobile next-generation networks (5G) and Internet of Things (IoT) technologies, will allow the connection of millions of devices with high bandwidth and minimal latency. In addition, Big Data technologies, together with Machine Learning (ML) and Deep Learning (DL) techniques, will allow the analysis and extraction of information from the data in a matter of seconds [14].

Among all these technologies, IoT has had a huge impact on agriculture, enabling the integration of communication capabilities to sensors and actuators [11, 23]. This translates into the possibility of using hardware devices in the final installations to communicate information related to the production environment, such as the pH of the water, the level of fertilizers, or the level of luminosity. This information from the sensors is sent to a local server or a server in the cloud. In general, in order to achieve low latency responses, part of the data processing is performed in compute nodes close to the sensors following the Edge Computing (EC) paradigm [21]. In contrast, the rest of the information is stored in the cloud databases to be analyzed later. Once the analysis is completed, IoT technologies also allow actuators to receive

✉ Ángel Luis Perales Gómez
angelluis.perales@um.es

Pedro E. López-de-Teruel
pedroe@um.es

Alberto Ruiz
aruiz@um.es

Ginés García-Mateos
ginesgm@um.es

Gregorio Bernabé García
gbernabe@um.es

Félix J. García Clemente
fgarcia@um.es

¹ Departamento de Ingeniería y Tecnología de Computadores, University of Murcia, Murcia 30100, Spain

² Departamento de Informática y Sistemas, University of Murcia, Murcia 30100, Spain

commands to take corrective actions on the system. This communication is usually performed through wireless channels, and it is frequent that IoT devices form Wireless Sensor Networks (WSN) using open standards.

In recent years, ML/DL technologies have gained prominence in the context of crop quality prediction [31]. These techniques are based on developing models capable of extracting information from the input data and continuously predicting the final product quality. The development of these models is usually divided into two distinct stages. The first stage is called training and is where the model learns the information underlying the data. In the second stage, called the test, the models are tested with previously unseen data to determine their performance. Thus, ML/DL can analyze information from agricultural sensors and improve decision-making tasks.

Despite the extensive use of IoT architectures on farms and in the field of precision agriculture, many of these architectures are aimed at monitoring variables. One of the biggest challenges today is integrating IoT architectures with ML/DL models that help us extract precise information to make the right decisions. In this way, the production of farms can be improved. The current solutions only monitor crops to extract statistics that can be used to improve future plantations. However, some unexpected events can be produced during the crop growth that can reduce the plantation production. Combining monitoring and corrective actions will minimize the effect of these unexpected events, and therefore maximizing the production. In the specific case of crop quality, another relevant challenge is the assessment of the product quality using information from diverse sensors such as environmental sensors or RGB cameras. For example, RGB cameras can give more insight into certain pests that perform a visual degradation of the crop color. In contrast, other sensors such as wind speed or pH water can give more information about chemical and physical information. Data aggregation and harmonization are complex tasks that must be tackled [6].

To overcome the aforementioned challenges, we present the following contributions:

- FARMIT, which is an IoT architecture designed for continuous crop quality assessment. The layers of FARMIT can be divided into three categories: physical, edge, and cloud. The goal of these layers is to gather information about the crop and analyze them. Based on the analysis, the architecture can take corrective actions to improve the quality of crops.
- A deployment of FARMIT in a real scenario in order to assess the quality of tomatoes under greenhouses. We show the experimental result obtained from this deployment, where FARMIT uses ML/DL techniques.

The remainder of this manuscript is structured as follows. Section 2 reviews the state of the art in terms of IoT architectures proposed for smart farming and precision agriculture. In addition, in this section, ML/DL techniques for crop prediction and crop quality are also reviewed. In Sect. 3, we present the IoT architecture to gather and store sensing information from farms. The scenario where we applied the previous architecture is presented in Sect. 4. In Sect. 5, the results regarding the crop quality forecasting and prediction are presented. Finally, the conclusions and future work are presented in Sect. 6.

2 Related work

In this section, we review the literature in the field of both IoT architectures in farms and precision agriculture and the usage of ML/DL techniques to evaluate crop quality.

2.1 IoT architectures for farms and precision agriculture

The IoT architectures combine different communication protocols, security mechanisms, and smart devices that are resource constrained [12]. Generally, these architectures are supported by the Fog [28] or Edge [21] computing paradigm. For example, the authors of [24] propose a novel and secure Cache Decision System following the Fog computing paradigm that operates in a wireless network focused on smart-buildings. In this sense, the Cache Decision System proposed enables a safer and more efficient environment for internet browsing and data management. In addition, new approaches have been proposed to improve the performance of such architectures. For example, the authors of [3] introduce a variant of the optimistic concurrency control protocol in cloud-fog environments. They probed that the proposed mechanism reduces the communication delay significantly and enables low-latency fog computing services of the IoT applications. In the security context, the authors of [18] propose a novel approach to generate and detect watermark in images with the goal of share image information in smart cities y a safety way. On the one hand, the generation method takes a gray image watermark that is encoded as watermark signal into the block Discrete Cosine Transform (DCT) component. On the other hand, the method to detect and extract the watermark is based on a cooperative Convolutional Neural Network (CNN).

In the agricultural context, the usage of IoT has expanded in the last years, and its use is well documented in the literature [20]. For example, the authors of [16] presented an IoT architecture for strawberry disease prediction especially designed for smart farms. The

architecture is capable of handling the collection, analysis, monitoring, and prediction of agricultural environment information. The authors also presented an IoT-hub network model that enables efficient data transfer between IoT devices. Several IoT-hubs can be deployed, and the communication is powered by LoRa technology. In addition, the IoT-hub communicates with the upper layers of the architecture using the oneM2M common platform. On top of the architecture, the authors presented a service capable of analyzing the data and predict strawberry infection. The authors of [25] presented a generic reference architecture for monitoring remote sensing in the field of precision agriculture. The work proposed a 7-layer architecture and discussed the technologies employed in each layer. To be specific, the layers are: sensor, link, encapsulation, middleware, configuration, management, and application layer. The authors also presented a use case based on a 24-hour real-time saffron cultivation surveillance system that relies on signal and image collection and preprocessing. The authors of [7] proposed another IoT architecture for smart farming called LoRaFarM. This architecture is low-cost, modular, and Long-Range Wide-Area Network-based IoT platform intended to manage generic farms in a highly customizable way. In addition, the platform was evaluated on a real farm in Italy. In this evaluation, the LoRaFarM was collecting environmental data for three months.

Apart from the previous solutions, the other two architectures for smart farming were presented in [4, 29]. A point in common between those two works is the usage of Fiware [5], which is an open-source initiative for context data management, facilitating smart solutions. On the one hand, the architecture presented in [4] integrates IoT, Edge Computing, Artificial Intelligence, and Blockchain technologies. In addition, this architecture was aimed to monitor the state of dairy cattle and feed grain in real-time. On the other hand, the work proposed by the authors of [29] presents a flexible platform capable of coping with soilless culture needs in full recirculation greenhouses using moderately saline water. The architecture is supported by three planes: local, edge, and cloud. The local plane interacts with crop devices to gather information. The edge plane is in charge of managing and monitoring the main tasks. Finally, the cloud plane performs the data analysis process. Both of the previous architectures were implemented in real scenarios.

All the works examined propose architectures for only collecting one type of sensor data. This means that they ignore the control opportunities that IoT technologies provide. In the end, this will cause a loss in crop production and, therefore, an economic loss. In this work, we present an architecture for data collection and, depending on the results of the analysis performed, it can even take corrective actions to improve the quality of the crop, minimizing

production losses. In contrast to previously discussed solutions, our architecture allows to aggregate data from different sources such as traditional sensors that give data about physical and chemical properties and visual sensors such as RGB cameras that give data about physical appearance.

2.2 Machine learning and deep learning to evaluate crop quality

The usage of ML/DL techniques in agriculture has been widely explored [19]. To be specific, crop management, livestock management, water management, and soil management are the most prominent areas where ML/DL are applied. Among them, we are interested in crop management, where we can identify the following activities: yield prediction, disease detection, weed detection, crop quality, and species recognition.

In this context, crop quality is the subfield in charge of estimating the final quality of crops, and it is closely related to disease detection. The importance of this field is critical since the price and the competitiveness of companies depend on the quality of their products. In this sense, the authors of [30] presented a study focused on the detection and classification of common types of foreign matter embedded inside cotton lint. During the study, a short wave near-infrared hyperspectral imaging system was used. The authors of [13] presented a study to differentiate between deciduous-calyx pear (DCF) and persistent-calyx pear (PCF). In the same direction that previous authors, a non-destructive hyperspectral imaging technique was used. The authors stated that PCF and DCF could be differentiated using the model proposed, which is based on Support Vector Machines (SVM). The final accuracy achieved was 93.3% for DCF and 96.7% for PCF. Another work to determine the quality was proposed by the authors of [26]. The authors stated that the quality of the rice depends on the origin country. They conducted experiments to determine the geographical origin of rice. Specifically, it was determined using inductively coupled plasma mass spectrometry (ICP-MS) together with different classification methods. Random Forest and SVM were the techniques that achieved the best performance (96%). The conclusion was that the variation in non-essential element profiles in rice grain depends on the geographical origin.

To predict diseases, the authors of [8] presented a new image processing technique to detect thrips (Thysanoptera) on strawberry plants. SVM was used for the classification of parasites and the detection of thrips. Images taken by a mobile agricultural robot were the input data to the SVM. The images were taken at 80 cm distance and under good natural light conditions. In addition, the images needed to be converted from RGB to HSV color space. The yellow

rust wheat disease detection is studied in [1]. The authors presented a methodology for the timely detection of yellow rust disease. For this, the authors used reflectance spectrum and a classification algorithm at different yellow rust development stages. Using a selection of the top 5% significant spectral features, the authors achieved a true positive rate of 86%. In [9], a CNN approach was used for identifying plant disease, focusing only on plant leaves images. Several CNN architectures were tested, being the one based on VGG, the one that achieved the best success rate (99.53%). The model was trained with 87 848 images, and it was tested with 17 548 images.

Finally, in [22], an IoT architecture for smart farming that incorporates ML algorithms was presented. The ML method incorporated in the architecture is based on the PART classification technique, and it is able to predict crop productivity and drought.

Most of the works examined propose evaluating the quality of the crop or carrying out the detection of pests from the study of the crop images. This approach ignores other useful chemical information that is crucial to determine crop quality. As a result, considering only the images at specific moments of crop growth limits the development of predictive models. For example, to develop a model to predict the quality of a crop in the following week, it is necessary to provide the model with other features that provide information about the conditions in which the crop has grown, such as the pH water, wind speed or pesticides used. In our proposal, visual information is considered to evaluate the appearance of the crop. However, measures of quality, growth, and pest registration are also used to predict the quality of the crop in the near future.

3 FARMIT architecture

In this section, we describe the proposed IoT architecture to evaluate crop quality. The architecture, called FARMIT, was designed to be scalable and flexible. To accomplish these requirements, FARMIT makes use of both the cloud [27] and the edge [21] computing paradigm and it is powered by FIWARE [10]. The architecture presents three different layers: physical, edge, and cloud. The physical layer is the nearest to the activities carried out on the farm. This layer allows the data acquisition and transmission to the Farm Controller (FC) located in the edge layer. The FC receives data from the previous layer and transmits it to the cloud layer. In addition, the FC is responsible for controlling and managing the infrastructure. Finally, in the cloud layer, we find three tiers: data, analysis, and application. The data tier is in charge of receiving data and context information from FC and storing it in the cloud database (Cloud DB). The analysis tier is in charge of

processing information and extracting relevant metrics and features from the data. Another responsibility of this tier is to train a prediction model and integrate a decision maker based on the predictions. Finally, the application tier comprises multiple operational and business applications that can be developed over the analysis tier.

The FARMIT architecture works in two different phases: training and production. During the training phase, the architecture uses all the data gathered from sensors to train the ML/DL models in the analysis tier. In contrast, during the production phase, FARMIT is ready to evaluate data recently gathered, analyze them, and perform the proper action to correct any anomaly. Except for Decision Maker and Action Enforcement Module, the modules work similarly in both phases. In particular, these two modules mentioned above do not take place in the training phase.

A global view of the architecture is presented in Fig. 1. In the following sections, we detail each layer and the components that integrate them.

3.1 Physical layer

This layer is the closest to the activities carried out on the farm. It is mainly made up of sensors responsible for collecting data from different sources and executing specific actions using actuators. Both sensors and actuators are implemented in combination with resource-limited hardware, i.e., Microcontroller Units (MCU) [2]. MCUs are in charge, on the one hand, of obtaining information from the

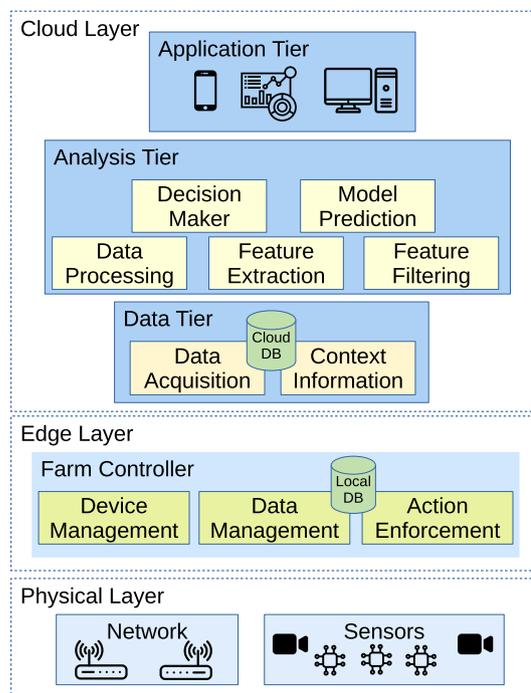


Fig. 1 Scheme of the proposed FARMIT architecture

sensors using the analog-digital converters. On the other hand, they interact with the actuators through digital-analog converters. In addition, to communicate with the sensors and actuators, the MCU can use different protocols designed for this purpose. It is usual to follow a master-slave communication scheme where the MCU initiates communication, and the sensor or actuator responds to the request. Among the protocols used in this area, we can find the SPI and I2C protocols [17].

There are a wide variety of sensors, from those that measure air quality to those that determine the pH level in the water. In general, it is common to find sensors that take measurements of temperature, ambient humidity, soil humidity, electrical conductivity, wind speed and direction, carbon dioxide, pH, light intensity, solar radiation, and atmospheric pressure, among others. Concerning actuators, we also find different types, but in general, they are related to sensors. It is common to find actuators that correct the deficiencies measured by the sensors. For example, if the soil humidity measured by a sensor is not adequate, an actuator will proceed to let more or less water pass to the crop as necessary. Regarding FARMIT architecture, it is focused on data aggregation, and it is not limited to certain types of sensors or actuators, but also FARMIT can integrate a wide range of these devices since the upper layer is in charge of managing them. In this sense, our proposal gives freedom when it comes to deploying the necessary sensors for a particular application.

Additionally, the devices deployed in this layer usually have a restriction in relation to energy consumption. Frequently, these devices are deployed in the open field where it is not possible to connect them to the electrical network. Therefore, it is necessary to equip these devices with batteries. Thus, the optimization of electricity consumption becomes a critical aspect to extend the life of batteries.

The other responsibility of this layer is to allow communication between the sensors/actuators and the FC. Specifically, we can find gateways that act as wireless access points for the infrastructure devices and route data packets. The need for these gateways is motivated by two fundamental points. The first is to homogenize the communication protocols, and the second is related to the optimization of the energy consumption of the sensors/actuators.

3.2 Edge layer

The FC is located at the edge and serves as an intermediary between the physical and cloud layers. The goal is to achieve low latency in communication with the sensors layer and carries out controlling and management tasks on physical devices. The FC consists of three modules: data management, device management, and control

management. The FC can be deployed in a physical or virtualized server in the farm facility, and each FARMIT deployment needs to be considered the current sensors and the future sensors that can be deployed.

3.2.1 Device management module

This module is in charge of managing the devices that are in the lower layer. In particular, this module manages the connection of new sensors/actuators in the FARMIT architecture and, moreover, ensures the correct operation of all the devices of the lower layers. When a new device is connected, it must communicate with the device management module to obtain its configuration. For example, a configuration parameter could be the interval in which samples must be collected by sensor devices. Also, network gateways get both routing and security configuration information from the device management module.

Once the device is configured, the device management module gives it a unique ID and records its activity in the local database. In this database, in addition to recording the data sent by the device, there is also information on the location of the device and its network configuration. Using this information, the device management module can carry out periodic checks to verify communication with the device is correctly configured, and information is being sent according to the established intervals.

Since FARMIT is an architecture powered by FIWARE, this module is implemented using the proper IoT Agent, which is a Backend Device Management Generic Enabler. In particular, this component behaves as the gateway to route the information to the upper layer using specific topics defined in the MQTT server.

3.2.2 Data management module

This module is in charge of managing and storing the data from sensors. The data model used by FARMIT is provided by the FIWARE project, i.e., the agrifood data model. When the devices collect data from the physical world, these data are sent to the FC, which are received by the data management module that has mainly two functions. The first of these functions is to preprocess and store the data in the local database on the FC. The second function is to synchronize the FC database with the database in the cloud layer.

The data preprocessing allows removing unwanted data and noise filtering present in the data. On the one hand, the data from the sensors can represent a large volume of network traffic and may even contain data that is not relevant to the tasks to be performed in the cloud layer. This module allows us to select which data will be transferred to the cloud. Imagine that the FC receives a wide range of

data that are not required for the crop quality task, but it is essential to carry out other tasks such as sensors or crop tracking. In the control panel, the operators can define that this information must not be sent to the cloud. On the other hand, the physical layer sensors can introduce noise inherent to the technology used or present outliers. This module allows us to filter this noise so that the data that finally reach the cloud layer are useful for quality assessment tasks. For example, imagine that a certain temperature sensor sends extremely large values because of a malfunction, i.e., it is broken. In this case, the operator can establish basic rules to ignore such values until the sensor was replaced. It is worth mentioning that bot processes require the operator's intervention to define the action in the control panel.

Data synchronization is performed following a two-way scheme. This means that not only data from FC is sent to the cloud layer, but if there is a loss of information in the FC, the data management module can recover that information from the cloud database and store it in the local database located at the edge layer.

3.2.3 Action enforcement module

This module is in charge of monitoring and controlling the execution of commands in actuators. The FC is not only capable of monitoring the data sent by the sensors, but based on these measurements and decisions delivered by the cloud layer, it can autonomously take corrective actions. These commands end up reaching the actuators and producing an effect in the physical world that will ultimately correct the anomaly measured by the sensors.

On the one hand, the corrective actions can be directly applied by operators registered in the FARMIT application tier and propagated through the architecture until they reach the FC, where they are transmitted to the actuators. On the other hand, the local database stores the normal values, defining the range of values where measurements made by sensors must be enclosed. If any of these measurements are outside the range of normal measurements for each sensor, the FC can autonomously take action to correct this situation. A third way to perform a corrective action is based on the result reported by the Analysis tier. If this tier discovers any anomalies in the data, it can communicate with this module to take the proper corrective action.

To perform corrective actions, operators can define policies. They are stored both in the local database and in the database of the cloud layers. These policies are made up of an antecedent and a consequent. The antecedent establishes a condition based on data stored in the local database (e.g., sensor data) that, if fulfilled, will cause the

consequent to be evaluated. The consequent establishes the actions to take.

3.3 Data tier

Depending on the needs of the farm facility, it may be required to deploy two or more FC that will collect and manage their own data and devices. In many cases, the devices that each of the FC manages will be different. This layer receives data from the different FC deployed in farm facilities, managing and aggregating this data and managing the context information. This layer is made up of the Data Acquisition module and the Contextual Information module.

3.3.1 Data acquisition module

This module receives the information from the different FC deployed in the facility and is responsible for storing it in the database present in this layer (Cloud DB). The received information includes data from the sensors and includes data from the FC itself, such as its geographical position and operating statistics. Besides, each FC is identified with a unique ID that is assigned during its installation. This unique ID links the data stored in the database with each of the deployed FC. Among other functions, this allows that if there is a loss of information in the FC, it can recover the information by making a query to this layer and requesting the lost information.

3.3.2 Context information module

The main objective of this module is to harmonize and aggregate the data received by the data acquisition module and stored in the cloud database. The data received by this layer came from different sources. On the one hand, there are traditional sensors deployed in crop fields that measure temperature, humidity, or wind direction. On the other hand, the information in the form of images can also be received, either through fixed cameras or even cameras installed on drones. Additionally, the QA staff can provide operational information through applications deployed in the application tier. This information could complement IoT sensor measurements and, therefore, should be taken at different intervals and during different hours of the day. This module is in charge of extracting context information from all this data and making it available for later analysis in the upper layers.

This module is implemented using the Orion Context Broker provided by the FIWARE project. In particular, Orion Context Broker receives the data from the different sensors provided by the Data Acquisition module and

makes it available to the upper layer through the cloud database.

3.4 Analysis tier

This tier includes the necessary procedures to extract knowledge from the data tier. The main goal is to train an ML/DL model to serve the result on demand to the application tier. Most of the modules in this tier are oriented to process and curate the data to be used with ML/DL algorithms. Specifically, this layer comprises four modules: Data Processing, Feature Filtering, Feature Extraction, Prediction Model, and Decision Maker. The workflow begins by retrieving the data from the cloud database at the lower tier. This database contains both information from traditional sensors as well as data extracted from sensors capable of taking images (whether thermal, infrared, or RGB). This data goes through the Processing Module to adapt it to be used in the following modules. The Feature Filtering module removes those features that do not provide information to the model. Once the filtering is done, the Feature Extraction module extracts new features that provide more information to the model. Next, the Model Prediction module trains the model and provides the application tier with prediction results. Finally, the Decision Maker module compares the Model Prediction output with the corrective policies defined in the cloud database. If any policy is fulfilled, this module makes the proper decision to correct the crop quality deviation.

3.4.1 Data processing module

The first task that this module performs is to process all data stored in the cloud database. The data must be processed depending on its data type, i.e., an image needs to be processed differently from a traditional sensor such as a temperature sensor. For example, FARMIT supports the usage of visual and non-visual information. In the highly recommended case that RGB cameras are deployed in the plantation, one of the operations that can be performed is converting different color spaces. This operation allows extracting relevant features to the specified prediction problem in a later module. In contrast, all the data from traditional sensors are treated similarly by FARMIT architecture. In particular, one of the most popular ways of performing the processing is to aggregate data based on a statistic summary. The second task consists in scaling the continuous features present in the datasets. This task is extremely important since many ML/DL models perform better when the data is on the same scale. Finally, the third task carried out by this module consists in encoding the categorical variables following an adequate schema for the problem under study.

Additionally, under training mode, this module splits the dataset into training and test datasets and, moreover, offers two strategies for splitting. On the one hand, it offers the possibility of choosing these datasets using a random approach. In other words, the training and test datasets are generated from random samples of the original dataset. On the other hand, this module allows making the dataset division while preserving the temporal coherence of the data. In general, the second approach is especially useful when data is used as a time series.

3.4.2 Feature filtering module

This module is responsible for removing those features that do not provide information to the prediction model, degrading its performance in the worst case. One of the primary operations that this module uses to calculate which features do not provide information is to perform a study of the variance of each one. This study gives an estimation of how much the value of these features changes throughout the whole dataset. Those features that do not change are candidates to be removed by this module.

3.4.3 Feature extraction module

Once the features have been filtered, the main task of this module is to extract new relevant features. The primary technique used by this module is the use of different statistical metrics. This allows enriching the dataset with high-level features, extracting patterns from the raw data, and improving the performance of the model in terms of prediction.

Although each FARMIT implementation can significantly be different, we highly recommend including two sets of features. The first set is based on the statistical summary commented above. Among the metric we recommend are the mean, the minimum, the maximum, the standard deviation, and the range (defined as the difference between the maximum and minimum). In contrast, since crop growth is closely related to time, we recommend including some type of feature to encode the sense of time.

3.4.4 Model prediction module

This module is in charge of training the model using the training dataset and then evaluating the prediction model performance using the test dataset. Three tasks are defined to carry out the training phase: selection of the model, selection of hyper-parameters and their values, and training and fine-tuning of the model, as shown in Fig. 2.

The first task is to select an appropriate model to address the problem of crop quality prediction. In general, the models that work with sequences are good alternatives

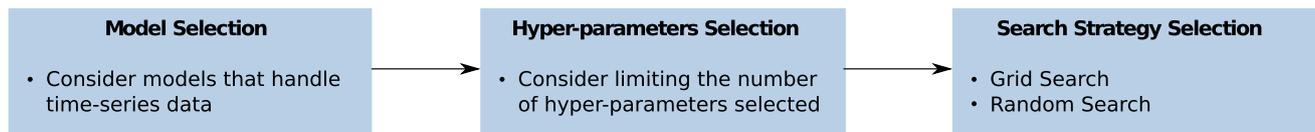


Fig. 2 Model Prediction Module phases in prediction mode

since the crop quality can be studied from the succession of the data given by the sensors over time. However, other models can be considered, achieving good results. For example, ML models such as SVM or Random Forests, or DL models as LSTM or RNN, could be considered. The second task is to define the hyper-parameters to be tested and their range. One consideration to keep in mind is that, in general, the greater the number of hyper-parameters to be tested, the longer the time in the fine-tuning phase of the model. Finally, the third task is to establish a search strategy for hyper-parameters. Among the most popular strategies, we highlight grid-search and random search. In general, the first one is used when a small range of hyper-parameters to be tested is selected. This will allow us to perform the search by testing all the defined hyper-parameters in an acceptable time. However, when a wide range of hyper-parameters to be tested is selected, it is recommended to use the random search strategy, although it will not test all possible combinations.

Once the model is trained, it is ready to make predictions on data that it has not previously seen. This mode of operation will be the most common in the FARMIT architecture. While the training mode should be done when the sensor data changes substantially, the prediction mode is used when a user wants to predict the quality of a specific crop.

3.4.5 Decision maker module

This module is in charge of deciding the corrective action to take in the case that any deviation is observed in the crop quality. To make decisions, this module consults the corrective policies previously defined in the cloud database. First, the prediction made by the trained model is obtained. Then, each policy is evaluated to determine if any of them is fulfilled. If, based on the predictions, any of the policy precedents are fulfilled, this module will evaluate the policy consequence to decide the action to take and thus be able to correct the deviations in crop quality. However, this module is not responsible for acting. Instead, that responsibility falls on the Action Enforcement module that receives the action to take and perform it.

Policies are based on the knowledge of experts and specify the actions to be taken. An example of a corrective policy is present in Algorithm 1.

Algorithm 1: Reduce water flow by 5% when sweetness is less than 7.5

```

if SWEETNESS < 7.5 then
  | REDUCE_AMOUNT_WATER(5%)
  
```

In this policy, we compare if the sweetness value is lower than 7.5, then the amount of water must be reduced by 5%. In the case that the antecedent is true, i.e., the prediction obtained for sweetness is lower than 7.5, then the corrective action is triggered and communicated to the system in order to be enforced by the corresponding IoT actuators.

In general, the antecedent of the policy is one of the variables predicted by the Model Prediction Module. In contrast, the consequent is an action to correct the anomaly.

3.5 Application tier

This tier is responsible for offering different operational and business applications to the operators. Thanks to the service-oriented interface that FARMIT exposes, these applications can be both desktop and mobile, as well as web applications. Regardless of technology, these applications communicate with FARMIT through REST services. Upon a specific request, the application tier communicates with the prediction module of the lower layer to obtain the results. These results will be returned to user applications as HTTP responses following a REST scheme.

4 Deployment in a tomato plantation

The FARMIT architecture was deployed in a tomato plantation in the south of Spain. The main objective was to evaluate and control the taste quality of cherry tomatoes. The plantation comprises three greenhouses that contained a certain number of tomato plants. In the physical layer, we deployed both traditional and visual sensors. The traditional sensors were in charge of measuring both physical and chemical properties, while visual sensors were intended to provide information about the appearance of the tomatoes. In particular, the traditional sensors deployed were: temperature, wind, rain, electrical conductivity, humidity, radiation, carbon dioxide, direction, and wind speed sensor. Regarding visual sensors, we deployed three

RGB cameras in several control points in the greenhouses. These cameras have to deal with different lighting conditions. For example, Fig. 3 shows as these lighting conditions vary depending on whether the images were taken during the day or at night. Another factor that can influence the brightness of the images obtained from the cameras is the meteorological state (cloudy, sunny, or rainy).

Additionally, in the three greenhouses where FARMIT was deployed, different operational applications for Human-Machine Interfaces (HMI) were deployed in the application tier, whose main objective was to allow the QA staff to input data. These data include necessary measurements performed by the quality department, the pests and defects that were affecting certain lines and greenhouses, and the activities carried out by the workers in the plantation. In relation to the pests that could affect the tomato, we highlight the whitefly, thrips, and tuta. Likewise, other defects that the tomato could present were registered, such as anomalies in its color, that the size was less than 22 mm or that the stem was less than 5 cm thick, among others.

All of these sensors were registered in the FC located in the edge layer through the Device Management Module. Once they were registered, they were allowed to send sensor data, which is managed by the Data Management Module. The data sent to this module was filtered and stored in the local database. This communication was performed using the IoT Agent provided by the FIWARE project.

Another data that was collected using applications was the scores obtained during the tastings tests. In particular, during the ML/DL training phase, a series of different properties of the tomato were scored by professional tasters and the QA department. The tastings were performed over a specific plantation, and each plantation can be evaluated by different tasting along the time. In each tasting, one or more tasters evaluated different samples collected from the plantation. Specifically, the measured tomato properties were: Brix Degrees, Maturity Index, Hardness, Sweetness, Acidity, and Tomato Smell. These six metrics, whose meaning is shown in Table 1, determined were used as tomato quality measures and, therefore, to label the data in the training phase of the ML/DL model.

5 Experimental results

This section details the specific operation of the different FARMIT modules to generate the predictive model. This process is only required initially when the model has not yet been generated, and the information from the different sensors is significant enough to determine the quality of the tomato. The objective of this predictive model is to evaluate tomato quality depending on the information from visual and non-visual sensors deployed in the greenhouses as well as the information collected by the QA department. This section is focused on the analysis tier and its

Fig. 3 Different light conditions in greenhouses

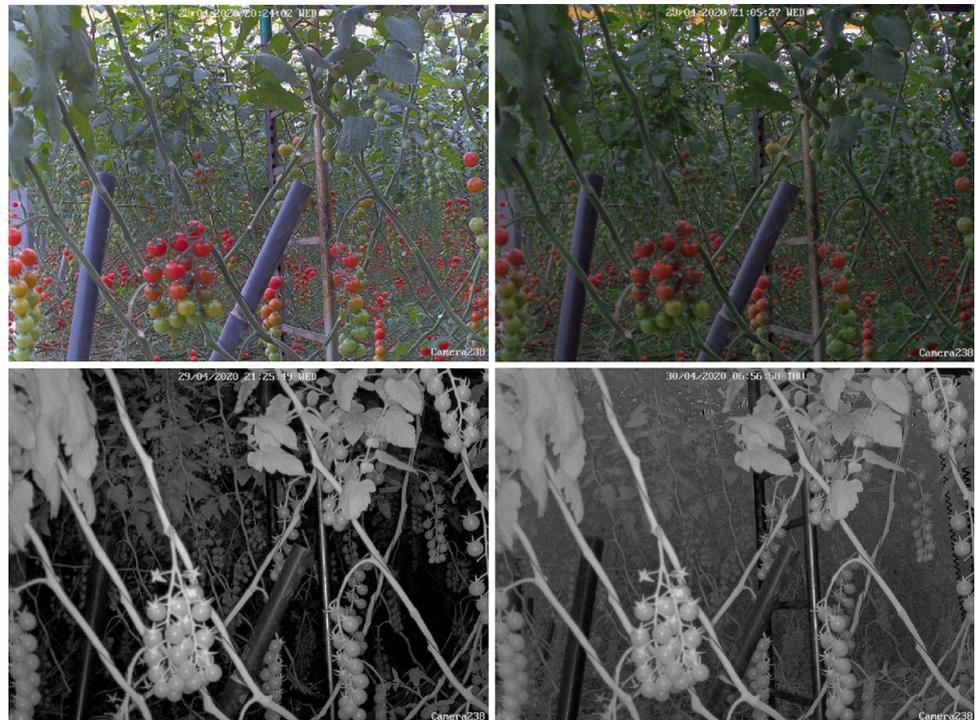


Table 1 Tomato properties measured and its description

Property	Description
Sweetness	The basic taste associated with aqueous sucrose solutions.
Acidity	The basic taste associated with aqueous solutions of citric acid.
Tomato smell	Aromas commonly associated with freshly picked fresh tomatoes, which are generally described as sweet (fructose), acid (citric acid), fruity, earthy, herbaceous, and ripe.
Hardness	The resistance presented by the tomato when biting it.
Brix degrees	Measures the concentration of soluble solids dissolved in a mixture, but due to the prevalence of sugars, pectins and organic amino acids in soluble solids of fruit and vegetable juices, it represents an estimate of the sugar content.
Madurity index	Value determined by the QA department based on the maturity of the tomato.

components since they are responsible for generating the model.

5.1 Data processing

Once the FC sent the information and it was stored in the cloud database, the next step performed by FARMIT was the data processing to adapt it to be used in the predictive model. Firstly, preliminary data processing was necessary because data were not collected at the same time intervals or under the same conditions. For example, the QA department weekly collected information about the growth of the crop (such as the number of leaves on a plant or the number of fruits on a plant) using a set of control plants for each season and each greenhouse, as shown in Table 2. In contrast, the values reported by the sensors in charge of collecting information about the water, such as electrical conductivity, water consumption, or pH level, were recorded for each day of the week as shown in Table 3. The other sensors, such as temperature, humidity, or wind speed and direction, were configured to communicate their measurements with different time intervals. While some communicated the measurements every 15 seconds, other sensors reported their measurements every 30 minutes.

Table 2 Crop growth measurements

Season	Week	Year	IdGreenhouse	Value	Sample
1	43	2019	1	15.0	1
1	43	2019	1	20.0	2
1	43	2019	1	16.0	3
1	43	2019	1	20.0	4
1	43	2019	1	16.0	5
1	43	2019	1	15.0	6

Season, Week, and Year indicate the season, week, and year in which the measure was taken. *Sample* indicates the sample measured. *IdGreenhouse* indicates the ID of the greenhouse where the sample was taken. Finally, *Value* indicates the value measured

Table 3 Values measured for pH

Season	Week	Year	IdGreenhouse	Value	Day of week
1	47	2019	1	3.69	1
1	47	2019	1	3.62	2
1	47	2019	1	3.58	3
1	47	2019	1	5.14	4
1	47	2019	1	2.98	5
1	47	2019	1	3.17	6

Season, Week, Year, and Day of week indicate the season, week, year, and day of week in which the pH was measured. *IdGreenhouse* indicates the ID of the greenhouse where the pH was measured. Finally, *Value* indicates the pH measured

This information was stored in a database table indicating the sensor to which they referred, a timestamp, and the reported value.

Table 4 shows the number of samples regarding its source. As can be seen, samples from sensors are the most numerous, followed by samples from images and samples related to quality. To relate all these variables, the data processing module groups them by the year and the week in which the measurements were taken. All the aforementioned data had this information available except for some sensors where a timestamp was available.

Table 4 Number of samples in the dataset regarding its source

Source sample	Number of samples
Task-related	2 521
Defect-related	1 495
Growth-related	6 579
Quality-related	17 248
Histogram-related	45 803
Sensor-related	8 497 937
Pest-related	355

Fortunately, the week number and the year could be extracted from the timestamp, allowing to group them with all the other database tables easily.

Additional work was carried out on integrating the information on the pests affecting the tomato crops, the tasks performed during the entire cultivation time, and the defects in the tomato crops during the entire cultivation time. These data were encoded following the One Hot Encoding (OHE) strategy, resulting in a dataset with 409 features divided into the categories shown in Table 5.

Additionally, this module created the time series to train the predictive model using two different approaches. The first approach generates time-series considering every week from the planting week to the week when the tasting was carried out. In other words, tasting number n included data for all the weeks between the week of planting and the week in which the tasting was performed. The second approach takes advantage of the fact that each plantation was exposed to several tastings tests. Therefore, this second approach generates the time series considering the weeks when the current and the last tastings were performed. In other words, if a tasting was carried out in week number 12 and another in week number 17, the generated time-series only contains the data between these two tastings tests. In both approaches, each of the generated time series was labeled with the mean of the tasters' scores.

Once all the data were grouped, the data processing module generated the training and test datasets. This partition could be done in two different ways. The first was to divide the data preserving temporal coherence between them. In contrast, the second consisted of selecting these datasets in a uniform random way, following a standard i.i.d. sampling scheme. Both approaches followed an 80/20 partition scheme. In other words, 80% of samples were used for training purposes, and 20% of samples were used for testing.

Table 5 Number of features in the dataset after the data processing step

Features	Number of features
Task-related	104
Defect-related	26
Growth-related	48
Quality-related	48
Histogram-related	120
Sensor-related	56
Pest-related	3
Others	4

The category *Others* includes: year, week, season, and IdGreenhouse

The last task of this module consisted of scaling the data so that all of them were on the same scale and could be used to train ML/DL models.

5.2 Feature filtering

This module removed those features that did not contribute information to the model. To accomplish this task, a study of the variance of each of the features in the dataset was performed. Finally, those features whose values remained constant throughout the dataset were removed. Specifically, a total of 248 features were removed from the dataset. Additionally, the year, week, and season features were removed since they did not provide useful information to the model. Taking into account the previous features, the total number of features removed was 161, divided into the categories show in Table 6.

5.3 Feature extraction

This module extracted high-level features that could be useful for generating the predictive model. Specifically, three groups of features were extracted.

The first group of features was extracted from the information grouped by the data processing module. In order to relate the information from different sensors, the data were grouped by week and year. From this aggregation, the feature extraction module calculated a statistical summary. Specifically, the mean, the standard deviation, the variance, the median, the minimum, the maximum, the sum, and the range metrics were computed and added to the dataset.

The second group of features was extracted from the images captured by the RGB cameras. These images were converted to the Lab color space for two reasons. The first is that it allows separating the luminosity (the L channel), making it possible to independently study the color (channels a and b). The second reason is that Lab color

Table 6 Number of features removed in the feature filtering process

Features	Number of features
Task-related	97
Defect-related	21
Growth-related	0
Quality-related	0
Histogram-related	30
Sensor-related	10
Pest-related	0
Others	3

The category *Others* includes: year, week, and season

Table 7 Comparison between different results

	Sequential split	Random split
<i>Considering weeks between plantation and current tasting</i>	0.338	0.255
<i>Considering weeks between last tasting and current tasting</i>	0.284	0.186

space allows differentiating small color changes. Then, the histograms for each channel were calculated and used as new features for the predictive model.

The last group of features was drawn from the previously removed week feature. This feature was encoded in the form of a unit circle extracting two new features representing the sine and cosine. This time representation eases the learning of repetitive patterns.

5.4 Predictive model

This module trained the predictive model. In particular, a model based on a Random Forest (RF) regressor with 100 estimators was selected to test performance in our scenario. The selection of the RF model was made based on its explainability, which is a desirable property in tasks where the path followed to make the prediction is required. In particular, due to its tree representation, the decisions take by RF can be interpreted by the operator. The metric we used to evaluate performance was Mean Squared Error (MSE), defined in Equation 1. One of the most interesting properties of this metric is that it is sensitive to large errors. In other words, it is a good metric when there are potential outliers in the dataset. This is particularly interesting when considering data gathered by sensors that may introduce errors in the measurement or when the operator can introduce erroneous data in the database through HMI. The MSE computes the square difference between the ground truth, Y , and the predicted value, \hat{Y} , of each sample, i . Then, the result is divided by the number of total samples, n .

$$MSE = \frac{1}{n} \sum_{i=1}^n (Y_i - \hat{Y}_i)^2 \quad (1)$$

Since the data processing module allowed the selection of different approaches to generate the dataset, this resulted in a total of four possible combinations shown in Table 5 along with the error obtained in the test dataset. Despite the different dataset generation approaches, all of them generated a training dataset with 80% of the total samples and a test dataset with 20% of the samples.

The best result is reached when a random split and weeks between the last and the current tasting is considered. Specifically, this approach achieved an MSE of 0.186. The approach considering all the weeks between the tasting and the plantation and using a random split achieved the second-best result with an MSE of 0.255. The third best result was obtained by considering weeks

between the last and the current tasting and following a sequential split, achieving an MSE of 0.284. Finally, the worst result was achieved considering all the weeks between the plantation and the current tasting and following a sequential split. In particular, this approach reached an MSE of 0.338.

Although the approach combining random split and weeks between the last and the current tasting achieved the best result, it is not always a realistic approach. The main limitation is that we are facing a time-series task, and we need to take into account several considerations. The random split is not desirable because each sample is a sequence of data weeks, and part of these data weeks can be seen during the training. Besides, the approaches considering weeks between the plantation and the current tasting are risky in time-series sequence creation. Imagine that different tastings were performed over the same plantation, which is frequent. In this case, all time-series sequences related to this plantation must be included in the train or the test dataset but not distributed in both datasets. This is because future sample sequences will include data from past weeks that will be seen during the training process. Due to the drawbacks previously commented, we recommend using the approach considering weeks between the last tasting and the current tasting and following a sequential split scheme (Table 7).

To illustrate the performance on the test dataset achieved by the model, Fig. 4 shows the results for eight samples. The blue bars indicate the value received by the tasting tests, while the orange bars indicate the values predicted by the model. We can conclude that the six variables determined in a tasting test are predicted quite accurately with very few exceptions. We can see that sample number 0 is where there is a more significant deviation regarding the Maturity Index and hardness label. In addition, Acidity is the label that presents the highest relative deviation from the ground truth, resulting in the label that gets the worst performance. However, it can be seen that FARMIT achieved acceptable performance. For this reason, we consider that the usage of data from different sources such as those related to quality, growth, images, and traditional sensor, together with information regarding pests, defects, and tasks, provide valuable information when determining the values of the tastings.

Besides, we carried out another experiment to demonstrate that the approach combining features from different sources outperforms the approach using standard sensors.

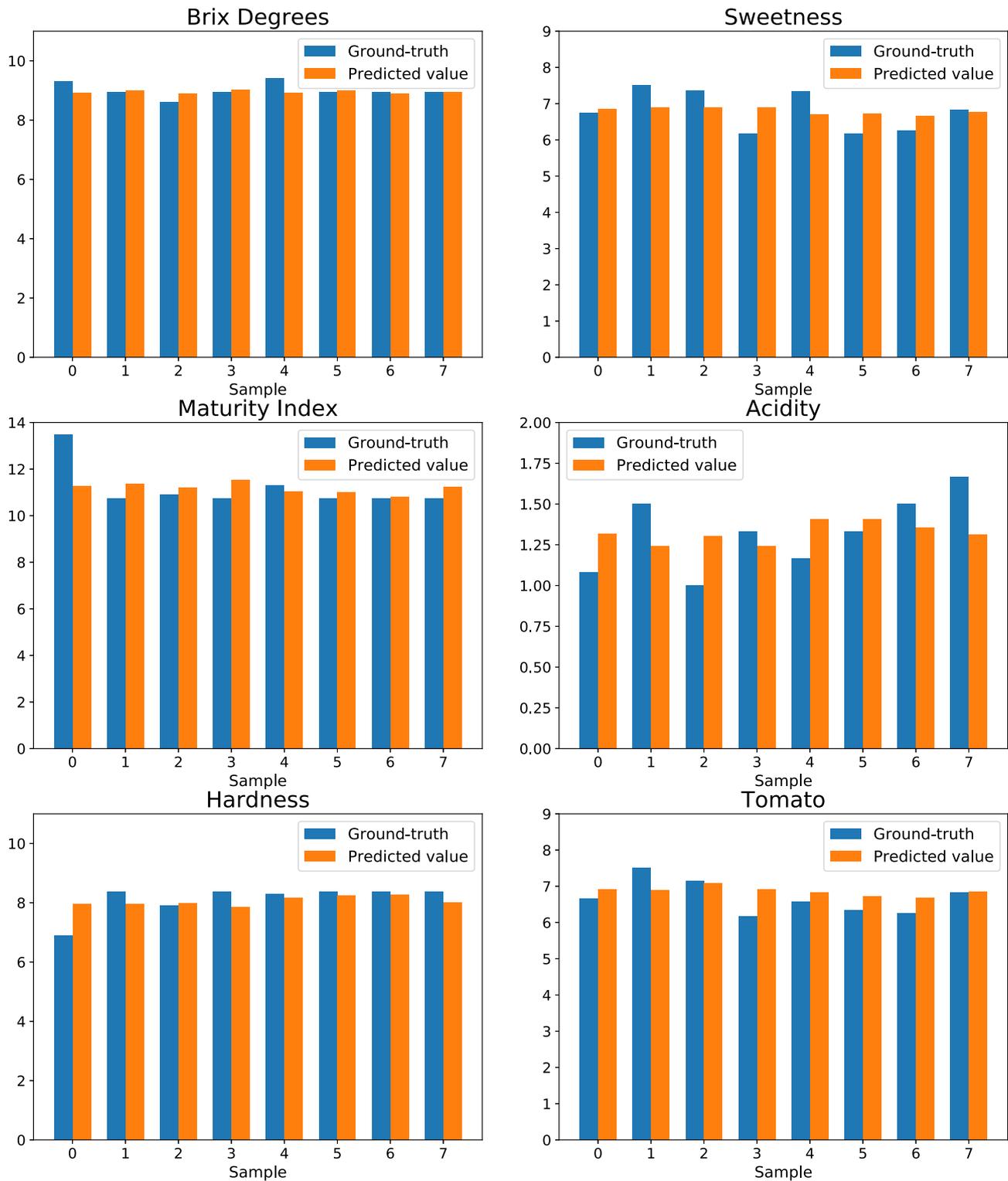


Fig. 4 Error in eight samples from the test dataset

To do this, we trained another RF model considering only the samples coming from the traditional sensors. Both the model trained using all features and the model using only traditional sensor features were trained using a sequential

split and week data between the last and the current tasting. Finally, we compute the percentage error of each predicted sample with respect to the ground truth. Table 8 shows the mean of the percentage error computed.

Table 8 Comparison between using all features and using only features from standard sensors

Approach	Percentage error (%)
Using all features	6.59
Using features from sensors	6.71

In conclusion, the approach that aggregates different sources achieved a better performance than the approach that only uses features from traditional sensors. This demonstrates that to carry out proper corrective actions, it is necessary to consider information from different sources.

5.5 Decision maker

Once the predictive model evaluates a sample, the output is examined to carry out the proper action. In our specific scenario, and taking into account the specific variables measured, the operators defined several policies based on the experts' knowledge shown in Algorithm 2

Algorithm 2: Decision maker algorithm

```

if
  SWEETNESS < 7.5 || TOMATO_SMELL < 6.5
  then
  | REDUCE_AMOUNT_WATER(5%)
else if ACIDITY != 1.7 || BRIX_DEGREE > 8
  then
  | REDUCE_AMOUNT_WATER(10%)
else if BRIX_DEGREE < 6 then
  | INCREASE_AMOUNT_WATER(5%)
if HARDNESS < 6 then
  | INCREASE_CONDUCTIVITY(15%)

```

The QA department concluded that when the sweetness is less than 7.5, the water reduction helps to correct the sweetness and back it to desirable levels. In addition, the reduction of water also helps to improve the acidity when it is not equal to 1.7, the Brix degrees when they are higher than 8, and the tomato smell when it is lower than 6.5. In contrast, the increase of water can increment the Brix degrees when they are lower than 6. Finally, the QA department probed that increasing the electrical conductivity of water when the hardness is lower than 6 can increase the aforementioned parameter.

6 Conclusions and future work

More and more farms and agricultural fields are automating their processes to improve their productivity. This automation, in most cases, is achieved by means of sensors

that measure different variables and actuators that perform actions in the physical world, and therefore allow us to correct deviations in the system.

In this work, we proposed a novel three-layer architecture called FARMIT that uses both IoT and ML/DL technologies to carry out a continuous assessment of the crop quality using data from different sources. The architecture provides necessary mechanisms to analyze aggregated data, extract information from it and recommend actions to correct quality deficiencies. For this purpose, operators can define corrective policies that trigger actions when a certain parameter is outside its range. Additionally, we have deployed the architecture in a tomato plantation with both sensors that obtain visual information (RGB cameras) and non-visual information, i.e., temperature, wind direction, or pH. From these data, together with the data on pests, defects and tasks carried out on the crop, an evaluation of the tomato quality was performed. For this, a Random Forest model was used to assess the crop quality, obtaining results very close to those determined by a professional taster. Besides, we conducted another experiment to compare the performance of our proposal that considers data from different sources and a traditional solutions that only consider data from sensors. In this sense, our proposal achieved a lower percentage error (6.59%) than a traditional solution (6.71%).

As future work, we consider the inclusion of new types of information sources, such as aerial images taken from drones. This will allow us to obtain graphical information on the entire plantation without installing a large number of cameras. Additionally, we plan to test DL models that improve the results we have obtained in this work.

Acknowledgements This work has been funded by Spanish Ministry of Science, Innovation and Universities, State Research Agency (AEI), FEDER funds, under Grants RTI2018-095855-B-I00 and RTI2018-098156-B-C53.

Author contributions Conceptualization, ÁLPG and FJGC; Data curation, ÁLPG, PELT and ARG; Formal analysis, ÁLPG, PELT, ARG and GGM; funding acquisition, GBG and FJGC; investigation, ÁLPG, PELT, ARG, GGM, GBG and FJGC; methodology, ÁLPG, PELT and ARG; project administration, GBG and FJGC; software, ÁLPG, PELT and ARG; supervision, GGM, GBG and FJGC; validation, GGM and FJGC; writing—original draft, ÁLPG, PELT, ARG and FJGC; writing—review and editing, GGM, GBG and FJGC.

Funding Open Access funding provided thanks to the CRUE-CSIC agreement with Springer Nature. This work has been funded by Spanish Ministry of Science, Innovation and Universities, State Research Agency (AEI), FEDER funds, under Grants RTI2018-095855-B-I00 and RTI2018-098156-B-C53.

Data availability The data presented in this study are not publicly available due to competitive reasons.

Declaration

Conflict of interest The authors declare that they have no conflict of interest.

Ethical approval Not applicable.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

References

- Aharoni, R., Klymiuk, V., Sarusi, B., Young, S., Fahima, T., Fishbain, B., Kandler, S.: Spectral light-reflection data dimensionality reduction for timely detection of yellow rust. *Precis. Agric.* **22**(1), 267–286 (2021)
- Al-Kofahi, M.M., Al-Shorman, M.Y., Al-Kofahi, O.M.: Toward energy efficient microcontrollers and internet-of-things systems. *Comput. Electr. Eng.* **79**, 106457 (2019)
- Al-Qerem, A., Alauthman, M., Almomani, A., Gupta, B.: Iot transaction processing through cooperative concurrency control on fog-cloud computing environment. *Soft Comput.* **24**(8), 5695–5711 (2020)
- Alonso, R.S., Sittón-Candanedo, I., García, Ó., Prieto, J., Rodríguez-González, S.: An intelligent edge-iot platform for monitoring livestock and crops in a dairy farming scenario. *Ad Hoc Netw.* **98**, 102047 (2020)
- Araujo, V., Mitra, K., Saguna, S., Åhlund, C.: Performance evaluation of fiware: a cloud-based iot platform for smart cities. *J. Parallel Distrib. Comput.* **132**, 250–261 (2019)
- Avazpour, I., Grundy, J., Zhu, L.: Engineering complex data integration, harmonization and visualization systems. *J. Indus. Inform. Integr.* **16**, 100103 (2019)
- Codeluppi, G., Cilfone, A., Davoli, L., Ferrari, G.: Lorafarm: A lorawan-based smart farming modular iot architecture. *Sensors* **20**(7), 2028 (2020)
- Ebrahimi, M., Khoshtaghaza, M.H., Minaei, S., Jamshidi, B.: Vision-based pest detection based on svm classification method. *Comput. Electron. Agric.* **137**, 52–58 (2017)
- Ferentinos, K.P.: Deep learning models for plant disease detection and diagnosis. *Comput. Electron. Agric.* **145**, 311–318 (2018)
- Fiware: The open source platform for our smart digital future. <https://www.fiware.org/>. Accessed 14 June 2021
- García, C.G., Meana-Llorián, D., Lovelle, J.M.C., et al.: A review about smart objects, sensors, and actuators. *Int. J. Interact. Multimed. Artif. Intell.* **4**(3) (2017)
- Gupta, B.B., Quamara, M.: An overview of internet of things (iot): Architectural aspects, challenges, and protocols. *Concurr. Comput. Pract. Exp.* **32**(21), e4946 (2020)
- Hu, H., Pan, L., Sun, K., Tu, S., Sun, Y., Wei, Y., Tu, K.: Differentiation of deciduous-calyx and persistent-calyx pears using hyperspectral reflectance imaging and multivariate analysis. *Comput. Electron. Agric.* **137**, 150–156 (2017)
- Kamilaris, A., Kartakoullis, A., Prenafeta-Boldú, F.X.: A review on the practice of big data analysis in agriculture. *Comput. Electron. Agric.* **143**, 23–37 (2017)
- Khanna, A., Kaur, S.: Evolution of internet of things (iot) and its significant impact in the field of precision agriculture. *Comput. Electron. Agric.* **157**, 218–231 (2019)
- Kim, S., Lee, M., Shin, C.: Iot-based strawberry disease prediction system for smart farming. *Sensors* **18**(11), 4051 (2018)
- Leens, F.: An introduction to i 2 c and spi protocols. *IEEE Instrum. Meas. Magaz.* **12**(1), 8–13 (2009)
- Li, D., Deng, L., Gupta, B.B., Wang, H., Choi, C.: A novel cnn based security guaranteed image watermarking generation scenario for smart city applications. *Inform. Sci.* **479**, 432–447 (2019)
- Liakos, K.G., Busato, P., Moshou, D., Pearson, S., Bochtis, D.: Machine learning in agriculture: A review. *Sensors* **18**(8), 2674 (2018)
- Mekala, M.S., Viswanathan, P.: A survey: smart agriculture iot with cloud computing. In: 2017 International conference on microelectronic devices, circuits and systems (ICMDCS), pp. 1–7. IEEE (2017)
- Morabito, R., Cozzolino, V., Ding, A.Y., Beijar, N., Ott, J.: Consolidate iot edge computing with lightweight virtualization. *IEEE Network* **32**(1), 102–111 (2018)
- Rezk, N.G., Hemdan, E.E.D., Attia, A.F., El-Sayed, A., El-Rashidy, M.A.: An efficient iot-based smart farming system using machine learning algorithms. *Multimed. Tools Appl.* **80**(1), 773–797 (2021)
- Shafique, K., Khawaja, B.A., Sabir, F., Qazi, S., Mustaqim, M.: Internet of things (iot) for next-generation smart systems: a review of current challenges, future trends and prospects for emerging 5g-iot scenarios. *IEEE Access* **8**, 23022–23040 (2020)
- Stergiou, C.L., Psannis, K.E., Gupta, B.B.: Iot-based big data secure management in the fog over a 6g wireless network. *IEEE Internet Things J.* (2020)
- Triantafyllou, A., Sarigiannidis, P., Bibi, S.: Precision agriculture: a remote sensing monitoring system architecture. *Information* **10**(11), 348 (2019)
- Villafañe, R., Hidalgo, M., Piccoli, A., Marchevsky, E., Pellerano, R.: Non-essential element concentrations in brown grain rice: assessment by advanced data mining techniques. *Environ. Sci. Pollut. Res.* **25**(22), 21362–21367 (2018)
- Wang, L., Von Laszewski, G., Younge, A., He, X., Kunze, M., Tao, J., Fu, C.: Cloud computing: a perspective study. *New Generat. Comput.* **28**(2), 137–146 (2010)
- Yi, S., Hao, Z., Qin, Z., Li, Q.: Fog computing: platform and applications. In: 2015 Third IEEE workshop on hot topics in web systems and technologies (HotWeb), pp. 73–78. IEEE (2015)
- Zamora-Izquierdo, M.A., Santa, J., Martínez, J.A., Martínez, V., Skarmeta, A.F.: Smart farming iot platform based on edge and cloud computing. *Biosyst. Eng.* **177**, 4–17 (2019)
- Zhang, M., Li, C., Yang, F.: Classification of foreign matter embedded inside cotton lint using short wave infrared (swir) hyperspectral transmittance imaging. *Comput. Electron. Agric.* **139**, 75–90 (2017)
- Zhu, N., Liu, X., Liu, Z., Hu, K., Wang, Y., Tan, J., Huang, M., Zhu, Q., Ji, X., Jiang, Y., et al.: Deep learning for smart agriculture: concepts, tools, applications, and opportunities. *Int. J. Agric. Biol. Eng.* **11**(4), 32–44 (2018)

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Ángel Luis Perales Gómez received the Ph.D. degree in Computer Science from the University of Murcia (Spain). He is currently a postdoctoral researcher with the Department of Computer Engineering, University of Murcia. His research interests include deep learning, cybersecurity of industrial control systems, and the security of distributed communication networks.



Gregorio Bernabé García received the Ph.D. degree in Computer Science from the University of Murcia (Spain) in 2004. In 1998, he joined the Computer Engineering Department of the University of Murcia and became an Associate Professor in May 2004. He has developed several courses on Computer Structure and Computer Architecture. His current research interests include the development of optimizations to improve the performance of

multimedia and medical applications in chip-multiprocessor (CMP) architectures. He has published more than 50 refereed papers in different journals and conferences.



Pedro E. López-de-Teruel was born in Lorca (Murcia, Spain), in 1970. He received his Ph.D. degree in Computer Science from the University of Murcia in 2003. At present, he is Associate Professor in the Computer Engineering and Technology Department at the same institution. His current research areas include computer vision, indoor localization systems, and activity recognition in industrial environments, as well as machine learning algorithms

and techniques applied to these fields.



Félix J. García Clemente received the Ph.D. degree in computer science from the University of Murcia, in 2006. He is currently an Associate Professor with the Department of Computer Engineering, University of Murcia. His research interests include cybersecurity and management of distributed communication networks. He is the coauthor of over 100 scientific publications and an active member on different national and international

research projects.



Alberto Ruiz received his PhD in Physics from the Complutense University of Madrid in 1990. He is currently an Associate Professor at the Department of Informatics and Systems at the University of Murcia. His research interests include computer vision, machine learning, and scientific computing. He is the coauthor of over 50 scientific publications and an active member on different research projects.



Ginés García-Mateos obtained his Degree of Engineer in Computer Science from the University of Murcia (Spain) in 1998, where he joined the research group in computer vision, image processing and pattern recognition. He obtained his Ph.D. in Computer Science in 2007, also in the University of Murcia. His main areas of research are divided into two main branches: the application of computer vision in biomedical images, and image process-

ing in the agronomic domain.