IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 19 August 2025, accepted 18 September 2025, date of publication 29 September 2025, date of current version 14 October 2025.

Digital Object Identifier 10.1109/ACCESS.2025.3615717

RESEARCH ARTICLE

IoT and Machine Learning for the Forecasting of Physiological Parameters of Crop Leaves

ARTURO BARRIGA [1], **JOSÉ A. BARRIGA** [1], **ADOLFO LOZANO-TELLO** [1], **MARÍA J. MOÑINO** [2], **AND PEDRO J. CLEMENTE** [1]

[1] Quercus Software Engineering Group, Departamento de Ingeniería Sistemas Informáticos y Telemáticos, Universidad de Extremadura, 10003 Cáceres, Spain
[2] Extremadura Scientific and Technological Research Centre (CICYTEX), 06187 Guadajira, Spain

Corresponding author: Arturo Barriga (arturobc@unex.es)

This work was supported in part by the TED2021-129194B-I00 Project funded by MICIU/AEI/10.13039/501100011033 and European Union NextGenerationEU/PRTR, in part by the 0100_TID4AGRO_4_E Project, co-financed by European Regional Development Fund (ERDF) through European Commission's INTERREG VI-A Spain-Portugal Program (POCTEP) 2021–2027; and in part by European Union (European Regional Development Fund), co-financed at 85%, and the Regional Government of Extremadura, with the Ministry of Finance as the Managing Authority under Grant GR24099. The work of Arturo Barriga was supported by a Pre-Doctoral Fellowship awarded by the Government of Extremadura and co-financed by European Social Fund Plus (ESF+) under Grant 593911006022023000186 (PD23034).

ABSTRACT In the current context of accelerating climate change, agriculture faces a pressing need for proactive solutions that can help maintain crop health and improve resource use efficiency. Physiological parameters of crop leaves, such as Leaf-Turgor Pressure (P_{leaf}) and Leaf Temperature (T_{leaf}), serve as valuable indicators of crop water stress and health, providing essential information for orchard management. However, although Internet of Things (IoT) systems equipped with sensors attached to crop leaves make it possible to monitor these parameters, they only provide real-time measurements, which are often insufficient to anticipate and mitigate adverse future conditions. Therefore, this study proposes a novel integration of IoT and machine learning technologies to enable the forecasting of physiological parameters of crop leaves. For this purpose, an experimental plot of Japanese plum trees has been monitored for five months using P_{leaf} and T_{leaf} sensors. Additionally, weather conditions were recorded. Using the data gathered, machine learning algorithms have been applied to train learning models for P_{leaf} and T_{leaf} forecasting. Results report Support Vector Regression as the best algorithm with R-squared values of 0.96 and 0.99 in the forecasting of P_{leaf} and T_{leaf}, respectively (one-week forecast horizon). In addition, a comprehensive digital twin software system integrating the forecasting models has been proposed. Thus, this study represents a significant breakthrough in proactive crop management, laying the groundwork for more sustainable and resilient farming practices.

INDEX TERMS Digital twins, Internet of Things, leaf temperature, leaf-turgor pressure, machine learning, time series forecasting.

I. INTRODUCTION

Agriculture is a fundamental sector to global food security and economic stability, employing nearly 27% of the global workforce [1]. However, this industry, which consumes approximately 70% of the world's freshwater supplies [2], is increasingly facing the effects of climate change, such as more extreme weather patterns, rising temperatures, and intensifying water scarcity in arid and semi-arid regions [2]. These environmental shifts threaten not only crop productivity but also the sustainability of existing agricultural ecosystems [2]. In this context, traditional agricultural practices, reactive in nature, are often inadequate to address such challenges, and there is a need for proactive approaches.

Physiological parameters of crop leaves are critical indicators of crop water stress and overall plant health, and provide valuable information for farm management. Some of the most relevant leaf physiological parameters are Leaf Temperature (T_{leaf}) [3] and Leaf-Turgor Pressure (P_{leaf}) [4], [5].

On the one hand, T_{leaf} refers to the temperature of the leaf surface that is exposed to the atmosphere [3]. This variable is determined by both the environmental conditions and the

The associate editor coordinating the review of this manuscript and approving it for publication was Wei Quan.

physiological structure and activities of the plant [3]. T_{leaf} is a key factor influencing healthy plant growth and plays a crucial role in understanding crop transpiration processes, variety improvement, and yield prediction [3]. In this sense, T_{leaf} has been used in the literature for a wide variety of purposes, from disease detection at leaf level [6] to the assessment of crop water stress [7], [8], [9].

On the other hand, turgor pressure, in general terms, is defined as the pressure greater than ambient atmospheric pressure found in living cells with walls, and is generated by the osmotic flow of water [4]. In plants, turgor pressure plays an important role in processes such as growth, organ movement, and flowering [4]. In this sense, P_{leaf} can be used for a variety of purposes, from studying stomatal actions and plant transpiration processes [5] to assessing the impact of soil supplements on crops [10]. One of the most valuable applications of this variable is the assessment of crop water stress. P_{leaf} has proven effective for this purpose in crops such as persimmon trees [11], maize plants [12], citrus trees [9], hedgerow olive trees [13], [14], [15], apple trees [16], and pepper plants [17], among others.

These physiological parameters can be measured by Internet of Things (IoT) systems that include wireless sensors attached to crop leaves [9], [11], providing farmers and agronomists with valuable information on plant condition. However, despite their benefits, these systems are generally limited to capturing real-time measurements that reflect only the current physiological state of the crop. This limitation represents an obstacle to the implementation of proactive management strategies in agriculture, capable of anticipating and mitigating undesirable future scenarios.

Recently, advances in machine learning have led to proposals that anticipate the future with high performance through time series forecasting in a variety of domains [18], [19], [20], [21], [22], including agriculture [23], [24]. Therefore, this work explores the combination of the IoT and machine learning technologies to enable the forecasting of leaf physiological parameters (i.e., T_{leaf} and P_{leaf}).

For this purpose, an experimental field of Japanese plum trees of *Talete*, *Angeleno*, and *Fortune* cultivars has been monitored for five months using P_{leaf} and T_{leaf} sensors. In addition, meteorological data (ambient temperature, relative humidity, wind speed and direction, solar radiation, and rainfall) have been recorded from a weather station located at a distance of one kilometer from the experimental field. From this data, different machine learning algorithms have been used to train the forecasting models. Results reveal that Support Vector Regression (SVR) is the most effective, achieving R-squared values of 0.96 and 0.99 for the forecasting of P_{leaf} and T_{leaf}, respectively (forecast horizon of one week).

Thus, the novel approach proposed in this work enables the future physiological state of plants to be known in advance with high performance, allowing farmers and agronomists to make timely, data-driven decisions about orchard management. This shift from reactive to proactive decision-making has the potential to substantially improve crop yield and quality, while also enhancing the resilience and sustainability of farming practices. For instance, it is recalled that P_{leaf} and T_{leaf} are well-established indicators of crop water stress (among other factors), which directly inform irrigation requirements and enable the application of precision irrigation techniques [9], [25], [26]. By forecasting these parameters, water usage can be optimized, ensuring that crops receive the right amount of water at the right time.

In addition, sensors attached to crop leaves often fail, as they are exposed to weather conditions, animal activities, leaf movement, and the growth of the crop itself [27]. In this context, the proposed approach may also enable sensors deployed in hostile environments, such as crop leaves, to be more resilient to errors (i.e., faulty or missing measurements could be recovered by predicting them, and sensor faults and anomalies could be detected if the measured values differ significantly from those expected), thus reducing the high costs associated with leaf-attached sensors in agriculture.

Furthermore, given that digital twin technology in agriculture is still in its early stages and far from realizing its full potential [28], [29], particularly in the realm of predictive digital twins [30], a comprehensive digital twin software architecture integrating the T_{leaf} and P_{leaf} forecasting models has been proposed. Therefore, this work not only proposes a set of learning models for forecasting important crop physiological parameters but also introduces a complete digital twin system, illustrating how these models can be applied to enable advanced predictive capabilities.

Taking the above into consideration, a list of Research Questions (RQ) to which this work aims to answer is defined:

- RQ1. To what extent is it possible to forecast leaf physiological parameters (i.e., T_{leaf} and P_{leaf}) in Japanese plum trees using machine learning and IoT technologies?
- RQ2. What is the performance of the learning models in forecasting T_{leaf} and P_{leaf} in Japanese plum trees with a seven-day forecast horizon?
- RQ3. Which machine learning algorithms are the most effective for training T_{leaf} and P_{leaf} forecasting models?
- RQ4. How could the T_{leaf} and P_{leaf} forecasting models be integrated into digital twin systems to support predictive capabilities?

The main contributions of the work are listed below:

- A novel approach to anticipate the future state of crops through the forecasting of T_{leaf} and P_{leaf}, which are essential physiological parameters for assessing crop status and guiding orchard management.
- A set of novel forecasting models able to predict T_{leaf} and P_{leaf} in Japanese plum trees with a forecast horizon of one week. The best models obtained an R-squared of 0.96 in forecasting P_{leaf}, and 0.99 in forecasting T_{leaf}.
- An innovative approach to achieve greater robustness to faults in leaf-attached sensors.

- A complete digital twin software architecture integrating the P_{leaf} and T_{leaf} forecasting models.

The remainder of the manuscript is structured as follows. Section II discusses related studies on time series forecasting using machine learning techniques. Section III presents the materials and methods used in this work. Section IV presents the results of the machine learning models. Section V proposes a complete digital twin software architecture integrating the P_{leaf} and T_{leaf} forecasting models. Section VI discusses the contributions and limitations of this study. Finally, Section VII concludes the paper and proposes future lines of research.

II. RELATED WORK

Although P_{leaf} and T_{leaf} have proven to be valuable indicators in agriculture, to the best of the authors' knowledge, no recent studies have explored the application of machine learning techniques for the forecasting of these physiological parameters. Consequently, this section reviews relevant applications of time series forecasting using machine learning. This provides an overview of the current state of the art in time series forecasting techniques and lays the groundwork for the proposal presented in this article.

In this context, time series forecasting can be categorized into univariate and multivariate approaches. Univariate methods predict future values based solely on past observations of the target variable, while multivariate approaches incorporate interactions with other variables to improve predictions [31]. The latter is more realistic, as real-world events are typically influenced by multiple factors (e.g., traffic levels depend on weather, fuel prices, and workdays). However, multivariate approaches are computationally more complex [31].

Regarding univariate approaches, Mudannayake et al. [22] explore the application of machine learning algorithms in forecasting municipal solid waste production. For this purpose, historical data from five cities located in Sri Lanka, the United States, and Australia are used, and algorithms such as Linear Regression (LR), Auto ARIMA, Light Gradient-Boosting Machine (LGBM), and Random Forest Regressor (RFR) are applied. Since only past values of solid waste production are used at prediction time to forecast future values, the problem is addressed from a univariate approach. The results of the experiments report LR and RFR as the best algorithms, with Mean Absolute Percentage Errors (MAPE) in forecasts with respect to actual values ranging from 8.03% to 36.89% depending on the municipality tested.

On the other hand, an example of a multivariate approach is Shokouhifar and Ranjbarimesan [20]. In this study, a deep learning model based on Long-Short Term Memory (LSTM) is proposed to predict blood donations and demands during the COVID-19 pandemic at the Tehran Blood Center, Iran. The model is trained to forecast the next week's blood demand and donations using, at prediction time, previous blood demand and donations (target variables), as well as previous COVID-19 infections and deaths. Tests of the proposed model report a 6.1% and 6.5% MAPE in forecasting the number of blood donations and demands, respectively.

Additionally, it is important to note that multivariate approaches can incorporate future data, such as weather forecasts from online services, as predictor variables at the time of prediction. In this sense, in Markovics and Mayer [21] several machine learning algorithms are compared for the day-ahead forecasting of photovoltaic power, including LR, Lasso, RFR, LGBM, XGBoost, and k-Nearest Neighbors Regression (KNNR), among others. For this purpose, two years of data from 16 different photovoltaic plants located in Hungary are collected, as well as meteorological data on many variables at these locations. The learning models are trained so that the next day's weather conditions are used at the prediction time to forecast the next day's photovoltaic power (i.e., multivariate approach). The results show that Kernel Ridge (KR) and Multilayer Perceptron (MLP) are the best-performing algorithms.

In this context, machine learning has also been extensively employed for forecasting problems in agriculture. In Van Klompenburg et al. [32], a systematic literature review on crop yield prediction using machine learning is conducted. Furthermore, machine learning can be applied in agriculture for purposes other than crop yield forecasting. In Goap et al. [23], a smart irrigation IoT system based on soil moisture prediction for the next few days is proposed. In order to develop their proposal, the authors collect data on soil and weather conditions for three weeks using sensors deployed in an agricultural field. Then, a learning model is trained using the SVR algorithm, whose output is given to a K-Means clustering model to optimize the forecast. The approach is multivariate, requiring the learning model to incorporate past soil conditions, past meteorological data, and future weather forecasts at the time of prediction in order to estimate soil moisture for the coming days. The authors obtain a 0.10 Mean Squared Error (MSE) and 0.96 R-squared in the prediction of soil moisture with respect to actual values.

In short, machine learning techniques have shown significant promise in addressing time series forecasting problems across diverse fields, including agriculture. While both univariate and multivariate approaches are commonly used, the latter is often more realistic due to its ability to incorporate interactions with multiple variables, making it particularly suitable for complex real-world scenarios. The reviewed studies highlight successful applications of machine learning in areas such as waste production, blood donation forecasting, and photovoltaic power generation. In agriculture, machine learning has been applied to problems ranging from crop yield prediction to soil moisture forecasting, demonstrating its potential. Despite these advancements, no studies have been identified that explore the forecasting of leaf physiological parameters, specifically P_{leaf} and T_{leaf}, using machine learning techniques, highlighting the novelty of this work.

III. MATERIAL AND METHODS

In this section, the materials and methods employed to train the forecasting models are presented. This includes a description of the experimental field, the gathered dataset, data pre-processing techniques, the machine learning algorithms employed for model training, the hyperparameter optimization process, and the performance metrics used.

A. EXPERIMENTAL PLOT CONTEXT AND RESULTING DATASET

The experimental plot was conducted during 2020 with late maturing Japanese plum cultivars *Angeleno*, *Talete*, and *Fortune*, located in the Extremadura Scientific and Technological Research Centre (CICYTEX) - La Orden in Badajoz (Spain). A set of P_{leaf} and T_{leaf} sensors was deployed on them on 13 May and removed on 5 October. Throughout this period, the devices were taking a measurement approximately every five minutes. In addition to the measured value, each reading also included the identifier of the sensor (*sensorId*), the exact time instant at which the measurement was taken (*timestamp*), as well as a binary error code indicating whether the measurement was considered correct or erroneous according to the experts (*errorCode*). Since the hardware and operational details of the deployed sensors were previously described in Barriga et al. [27], it is not considered of interest to elaborate on this aspect in this document.

TABLE 1. Dataset description.

Feature	Description	Units
sensorId	Unique sensor identifier	-
timestamp	Date and time of measurement	-
errorCode	Error code	-
P_{leaf}	Leaf-turgor pressure	kPa
T_{leaf}	Leaf temperature	ºC
T	Ambient air temperature	ºC
RH	Relative humidity of the air	%
SR	Solar radiation	W/m²
RFa	Rainfall	L/m²
WS	Wind speed	m/s
WD	Wind direction in degrees (0°–360°)	°

Finally, meteorological data were collected from a weather station located one kilometer from the plot. From this weather station, meteorological information including ambient temperature (T), relative humidity (RH), wind direction (WD), wind speed (WS), rainfall (RFa), and solar radiation (SR) was collected. Note that all the information presented constitutes the dataset used in this work for the training of the forecasting models, which is outlined in Table 1.

B. DATA PRE-PROCESSING TECHNIQUES

Before training the machine learning models, a thorough data pre-processing phase was conducted to enhance data quality and ensure compatibility with the algorithms. This stage involved transforming, cleaning, and organizing the dataset to improve its usability, ultimately contributing to the efficiency and effectiveness of the subsequent modeling steps [33], [34]. Several pre-processing techniques were applied in this study, as outlined below.

1) SUBSAMPLING

Subsampling is the process of selecting a representative subset of data from a larger dataset, often to reduce computational complexity [35]. In this study, the P_{leaf} and T_{leaf} sensors recorded a measurement approximately every five minutes. However, these leaf physiological parameters exhibit minimal variation over such short time intervals, resulting in very similar or even identical consecutive measurements. Therefore, subsampling was applied by selecting only the first reading of each hour for each sensor and discarding the subsequent readings within the same hour. This approach effectively reduced the size of the dataset while preserving the gathered information, as removed values were considered redundant.

2) NULL AND ERROR HANDLING

In agriculture, sensors are exposed to a hostile environment, such as weather conditions, animal activities, and crop growth and movement, making them susceptible to malfunctions [27]. These issues can result in erroneous readings or the absence of data, commonly referred to as null values. In this work, two distinct approaches were applied to handle missing and faulty sensor data.

- Exclusion of sensors with persistent errors: Sensors with frequent errors or missing data throughout the monitoring period were excluded from the dataset.
- Retention and cleaning of sensors with temporary failures: Some sensors recorded erroneous measurements at the beginning of the experimental period, but were subsequently relocated by the experts on the leaves, after which they began to function correctly. Similarly, other devices started to fail towards the end of the experimental period. In both cases, only the erroneous readings were discarded, while the valid data during the rest of the monitoring period were retained.

The application of these data cleaning strategies ensures that only reliable and continuous measurements are used for model training. After applying these cleaning procedures, 17 sensors deployed across 13 different trees remained in the dataset. Specifically, nine sensors were located on seven *Angeleno* plum trees, six sensors on four *Fortune* plum trees, and two sensors on two *Talete* plum trees. Thus, a sufficient and representative amount of data from the original dataset was preserved, enabling effective training and testing of the forecasting models.

3) MIN-MAX NORMALIZATION

In this study, min-max normalization has been applied to all numerical features. Min-max is a scaling technique that transforms numerical data into a specific range ([0,1] in this study) [36]. This transformation ensures that all features contribute equally to the learning process and prevents

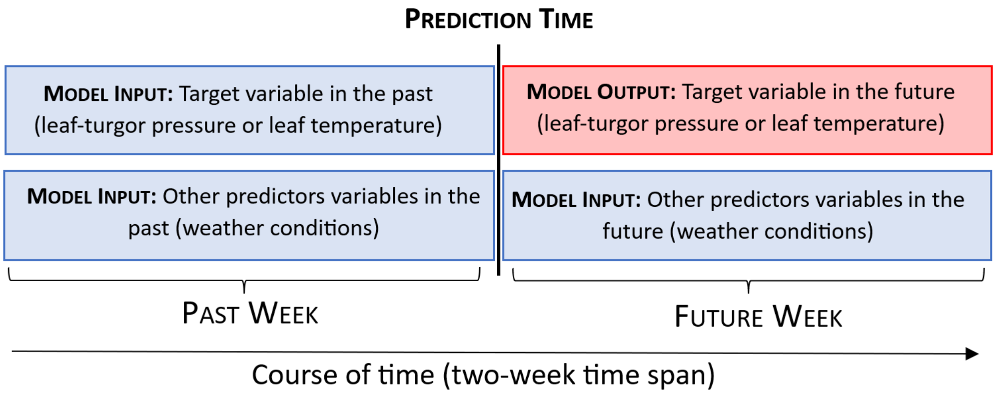


FIGURE 1. Diagram illustrating models' inputs (target variable in the past and other predictor variables both in the past and in the future) and models' output (target variable in the future). Target variable: variable to be predicted, i.e., P_{leaf} or T_{leaf}; other predictor variables: other variables that provide information for the prediction, i.e., weather conditions.

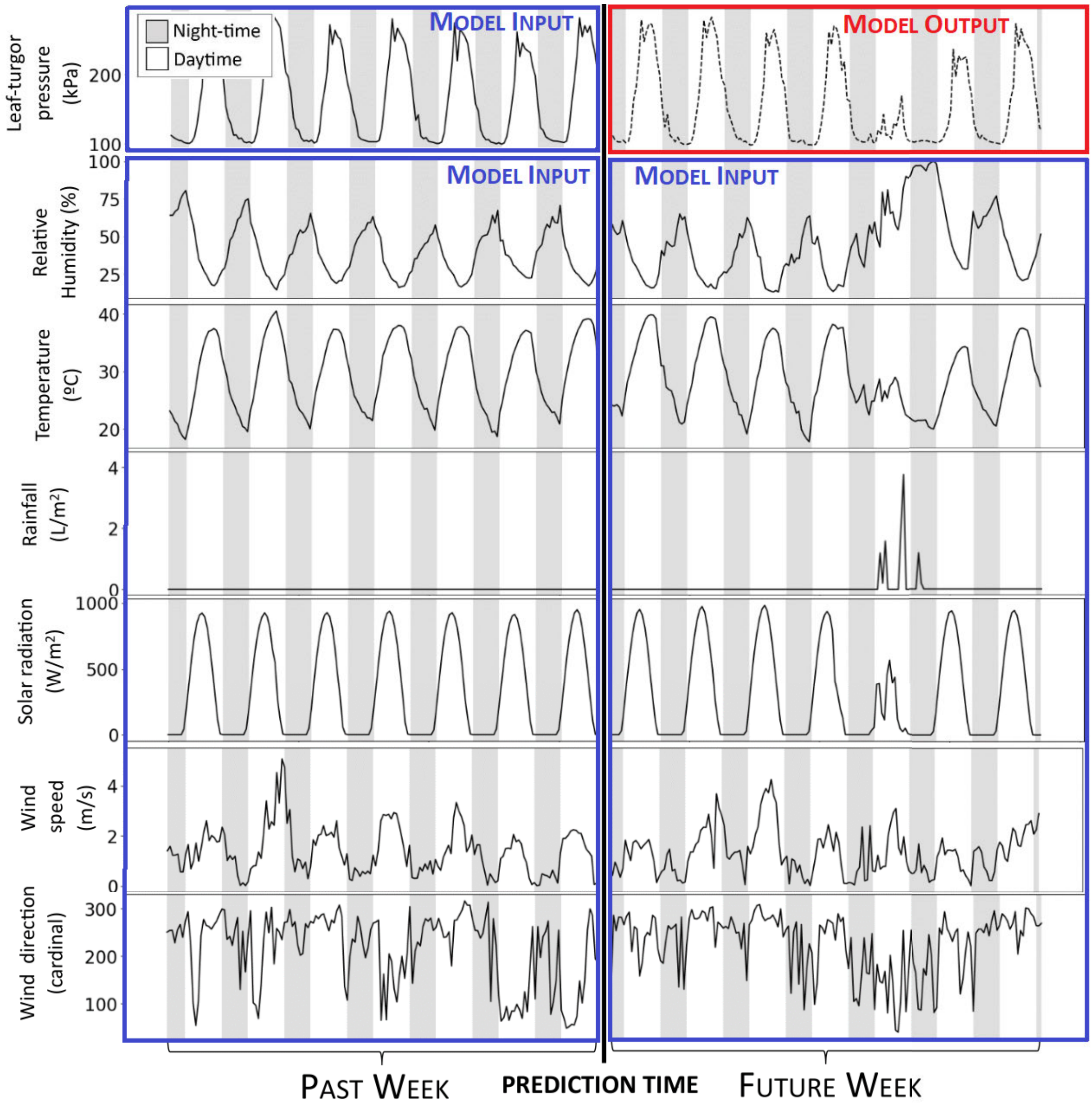


FIGURE 2. A real example of the input and output time series used to train the forecasting models. In this case, the target variable is the P_{leaf} of an *Angeleno* tree, and the time span is from the 10th of July 2020 to the 23rd of July 2020.

models from being biased toward variables with larger magnitudes.

Overall, these pre-processing techniques ensure that only complete and accurate data are provided to the machine learning algorithms, ultimately enhancing the reliability and predictive performance of the models.

C. MULTIVARIATE APPROACH: INPUTS AND OUTPUT OF FORECASTING MODELS

Learning models are trained with the aim of forecasting the target variable (T_{leaf} or P_{leaf}) with a forecast horizon of one week, i.e., seven days in advance from the prediction time. Therefore, the output of the models will be the forecasted values of the target variable for the future week.

Regarding the data used as input by the models to make their forecasts (i.e., predictor variables), a multivariate approach has been followed. First, the measured values of the target variable (T_{leaf} or P_{leaf}) in the past, i.e., during the last week, are used. Second, the weather conditions measured in the last week, together with the expected weather conditions for the following week, are also used as input data for the models. Figure 1 illustrates the inputs and outputs of the forecasting models.

On the one hand, providing the models with the behavior of the target variable (T_{leaf} or P_{leaf}) during the past as input is mandatory to predict the future variation of that variable. Each tree has different water stress, genetics, and, in general, different conditions that affect the physiological parameters of its leaves. Thus, the specific behavior of the target variable on that particular leaf during the last days must be taken into account in order to forecast its future variation.

On the other hand, providing the forecasting models with past and future weather conditions could be useful to achieve better performance. With this approach, models could learn more complex patterns and relationships between weather conditions and the leaves' physiological parameters, allowing them to better determine the future variation of the target variable. As the literature states, the leaf physiological parameters included in this work (i.e., P_{leaf} and T_{leaf}) are influenced by climatic conditions [3], [17]. In this context, it should be noted that weather conditions for the future week can be known at prediction time from weather forecasting services and agencies on the Internet such as the OpenWeather [37] or AccuWeather [38] APIs, and that several studies in the literature already follow this approach of using future weather conditions at the prediction time [21], [23].

Therefore, the approach adopted in this work is multivariate. Additional variables (past and future weather conditions) are used together with the target variable (P_{leaf} or T_{leaf}) in the past to predict the future variation of the target variable. Note that the meteorological variables used are solar radiation, relative humidity, air temperature, rainfall, wind speed, and wind direction, as shown in Table 1.

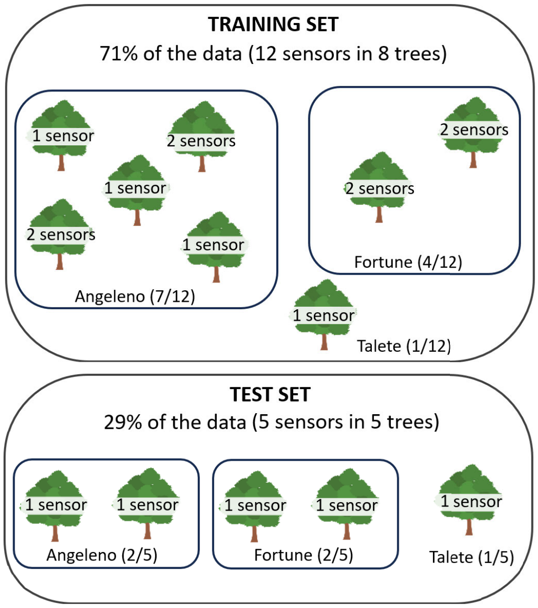


FIGURE 3. Training and test sets of forecasting models. Data from the same tree is never included in both sets. Icons from Flaticon.com have been used in this illustration.

Figure 2 provides a selected real-world example of the input and output time series used to train the learning models focused on forecasting P_{leaf} (since the selected target variable for this example is P_{leaf}). As explained above, the inputs are the measurements of the target variable in the past week together with the weather conditions in the past and future week, while the output are the expected values of the target variable in the future week.

The two-week time period shown in Figure 2 is from 10 July 2020 to 23 July 2020, both days inclusive and the P_{leaf} measurements were taken from an *Angeleno* tree. In this particular case, it can also be clearly seen that weather conditions directly affect the behavior of P_{leaf}, making it evident that in order to predict with high performance the future variation of this variable it is necessary to provide the models with the expected weather conditions for the coming week.

D. DATA SPLITTING: TRAINING AND TEST SETS

In order to properly train and measure the performance of forecasting models, it is necessary to split the data into two subsets. A commonly used method is to take a percentage of the records randomly which is used in the training phase and preserve the rest for the testing phase [39]. However, as each tree has different genetics, age, and, in general, different conditions, it is considered that the most appropriate option is to use a set of trees as a whole to train the forecasting models and use the remaining set of trees to test their performance. Measuring the performance of forecasting models with sensors placed on trees never seen during the training phase ensures that the performance obtained is the same as in a real-life environment and represents the actual ability of the models to generalize to unknown situations.

TABLE 2. Overview of the algorithms employed to train the P_{leaf} and T_{leaf} forecasting models.

Algorithm	Approach	Overview	Ref.
Linear Regression (LR)	Linear	LR predicts a target variable by finding the best-fitting linear relationship between input variables and the target	[40]
Ridge Regression (RR)	Linear	Ridge extends LR by adding to the loss function the sum of the squared values of the coefficients (L2 norm) multiplied by a predetermined parameter. This technique reduces overfitting	[41], [42]
k-Nearest Neighbors Regression (KNNR)	Neighbors	KNNR predicts a continuous output based on the mean of the target variable of its k-nearest neighbors	[43]
Support Vector Regression (SVR)	Support vector machines	SVR finds an optimal hyperplane in an n-dimensional space to predict the continuous output	[44]
CatBoost Regression (CatBoost)	Ensemble	CatBoost is a gradient boosting that employs an innovative algorithm highly effective in handling categorical data	[45]
LightGBM Regression (LGBM)	Ensemble	LGBM is an efficient implementation of gradient boosting	[46]

Figure 3 illustrates the partitioning of the sensors between the training and test sets. 71% of the data (12 sensors out of 17) is assigned to the training set and 29% (5 sensors out of 17) to the test set. As can be seen, not only is the data from each sensor completely assigned to one or the other subset, but also data from the same tree (as a tree can have attached more than one sensor) are never assigned to both subsets. Furthermore, the allocation of the sensors between the two subsets has been carried out taking into account a proportional distribution of the existing cultivars in the experimental plot (i.e. *Angeleno*, *Fortune* and *Talete*), avoiding a possible bias in this respect (see Figure 3).

E. ALGORITHMS USED TO TRAIN THE FORECASTING MODELS

A set of machine learning algorithms has been employed to train learning models focused on the forecasting of P_{leaf} and T_{leaf} in Japanese plum trees. These algorithms, selected for their ability to capture complex and diverse data patterns, are presented in Table 2. The implementation of the Python "Darts" [47] library has been used. Since the algorithms included are well-known, it has been considered unnecessary to include details of their inner workings in this article, as it would not provide any novelty.

At this point, it should be noted that neural networks were considered for model training in this research. However, the performance obtained by neural networks did not outperform some of the algorithms included in Table 2. Furthermore, neural networks involve substantial complexity in terms of topology design and training. Given the superior performance of some of the other algorithms, along with the additional methodological complexity associated with neural network design and training, it was deemed inappropriate to include neural networks in this manuscript.

F. HYPERPARAMETER TUNING

Hyperparameters are settings or configuration values used to control the learning process in machine learning algorithms. Unlike model parameters, which are learned from the data during the training process (e.g., weights in a neural network), hyperparameters are set before training begins and remain constant throughout the learning process [48]. In this regard, selecting appropriate hyperparameters is crucial, as they determine how well the model generalizes to unseen data and impact training efficiency and predictive performance [48].

TABLE 3. Hyperparameters tested for each algorithm in the grid search process.

Algorithm	Hyperparameters Tested
LR	-
RR	alpha: [0.05, 0.1, 0.5, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
KNNR	n_neighbors: [3, 5, 7, 11, 15, 21, 27, 33], metric: ["euclidean", "manhattan", "minkowski"], weights: ["uniform", "distance"]
SVR	C: [0.05, 0.1, 1, 2, 5, 7, 10], epsilon: [0.000001, 0.00001, 0.0001, 0.001, 0.01, 0.1, 1]
CatBoost	iterations: [500, 750, 1000], learning_rate: [0.01, 0.05, 0.1], depth: [2, 4, 6]
LGBM	num_leaves: [30, 50, 100], learning_rate: [0.01, 0.05, 0.1], n_estimators: [100, 300, 600]

In order to identify the optimal hyperparameters for each machine learning algorithm included, a grid search process was implemented. In this sense, grid search is a systematic approach that explores multiple combinations of hyperparameters, selecting the configuration that yields the best model performance based on a predefined evaluation metric (mean absolute error in the case of this work) [48], [49]. This exhaustive search process determines the most effective hyperparameter settings (among those tested) before model training.

The hyperparameters tested for each machine learning algorithm are detailed in Table 3. The results of the grid search process are presented along with the results of the forecasting models in Section IV.

G. METRICS USED TO MEASURE THE PERFORMANCE OF THE FORECASTING MODELS

To evaluate the performance of the forecasting models, several metrics were computed using the data from the test sensors: *Mean Absolute Error* (MAE) [50], which measures the average magnitude of errors in predictions, as illustrated in equation (1); *Mean Squared Error* (MSE), which calculates the average of the squared differences between actual and predicted values, giving more weight to

TABLE 4. Performance metrics obtained for each learning model in forecasting P_{leaf} using the test set.

Algorithm	MAE	MSE	RMSE	R-squared	Hyperparameters
Linear Regression (LR)	8.97	155.50	11.80	0.93	-
Ridge Regression (RR)	8.95	155.75	11.80	0.93	alpha = 2
k-Nearest Neighbors Regression (KNNR)	11.67	302.06	16.07	0.87	n_neighbors = 7, metric = “manhattan”, weights = “distance”
Support Vector Regression (SVR)	**5.92**	**95.80**	**9.00**	**0.96**	**C = 2, epsilon = 0.00001**
CatBoost Regression (CatBoost)	8.08	144.33	11.23	0.94	iterations = 750, learning_rate = 0.05, depth = 4
LightGBM Regression (LGBM)	7.07	125.96	10.56	0.95	num_leaves = 50, learning_rate = 0.05, n_estimators = 300

larger errors [50], see equation (2); *Root Mean Squared Error* (RMSE), which is the square root of MSE [50], calculated as depicted in equation (3); and *R-squared*, which represents the proportion of variance in the dependent variable explained by the model [51], computed as defined in equation (4).

$$\mathrm{MAE}(y, \hat{y}) = \sum_{i=1}^{n} \frac{|y_i - \hat{y}_i|}{n}, \tag{1}$$

$$\mathrm{MSE}(y, \hat{y}) = \sum_{i=1}^{n} \frac{(y_i - \hat{y}_i)^2}{n}, \tag{2}$$

$$\mathrm{RMSE}(y, \hat{y}) = \sqrt{\sum_{i=1}^{n} \frac{(y_i - \hat{y}_i)^2}{n}}, \tag{3}$$

$$\text{R-squared}(y, \hat{y}) = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2}, \tag{4}$$

where y represents the actual value to be predicted, $\hat{y}$ is the value predicted by the model, $\bar{y}$ is the mean of the actual values to be predicted, and n is the number of instances or predictions made.

Note that in Section IV the values provided for each performance metric (i.e., MAE, MSE, RMSE, and R-squared) of a model correspond to the weighted average of those obtained with each test sensor.

IV. RESULTS

Once the learning models have been trained, this section presents a comprehensive evaluation of their forecasting performance. First, the results for P_{leaf} are presented in Section IV-A. Subsequently, the performance of the models in forecasting T_{leaf} is examined in Section IV-B.

A. RESULTS OF P_{LEAF} FORECASTING MODELS

Table 4 shows the performance metrics obtained for each learning model in forecasting P_{leaf} using the set of test sensors. As presented in Table 4, the predictive performance of the models exhibits considerable variation. SVR yielded the highest performance, achieving the lowest MAE (5.92), MSE (95.80), and RMSE (9.00), along with the highest R-squared value (0.96). An R-squared value of 0.96 indicates that 96% of the variance in the forecasted P_{leaf} values is accounted for by the model's predictions, signifying a strong ability to capture the underlying patterns within the data.

By contrast, KNNR exhibited the weakest performance, yielding the highest errors (MAE = 11.67, RMSE = 16.07) and the lowest R-squared value (0.87), suggesting limited capability in modeling the complex relationships governing P_{leaf} dynamics. LR and RR exhibited comparable performance, both with an MAE close to 8.96 and an R-squared value of 0.93, indicating reasonably accurate yet less precise predictive capacity.

The gradient boosting models, CatBoost and LightGBM, performed better than LR and RR, with LightGBM achieving an MAE of 7.07 and an R-squared value of 0.95, ranking behind SVR.

In addition, to further evaluate the predictive performance of the best-performing model (i.e., SVR), Figure 4 presents four forecasts of P_{leaf}. In each plot, the black line represents the actual P_{leaf} values during both the observed and future weeks, while the orange line corresponds to the forecasted P_{leaf} values for the upcoming week. A dashed vertical line marks the prediction time, serving as a visual reference for the transition between observed data and predicted values. Furthermore, the background shading distinguishes between day and night, with gray denoting nighttime and white representing daytime. This distinction is particularly relevant in this work, as P_{leaf} exhibits diurnal fluctuations, typically increasing during daylight hours due to photosynthetic activity and decreasing at night.

At this point, it should be mentioned that the four forecasts shown in Figure 4 were randomly selected and were not chosen based on superior performance, ensuring a representative assessment of the model's capabilities. In this regard, the MAE for each forecast is included, providing an objective measure of the model's error in each specific case.

The forecasts illustrated in Figure 4 reveal that the SVR model effectively captures the dynamic behavior of P_{leaf} over time. Each forecast corresponds to different trees and/or time periods, leading to significant variability in P_{leaf} behavior. It is important to recall that P_{leaf} is a physiological indicator

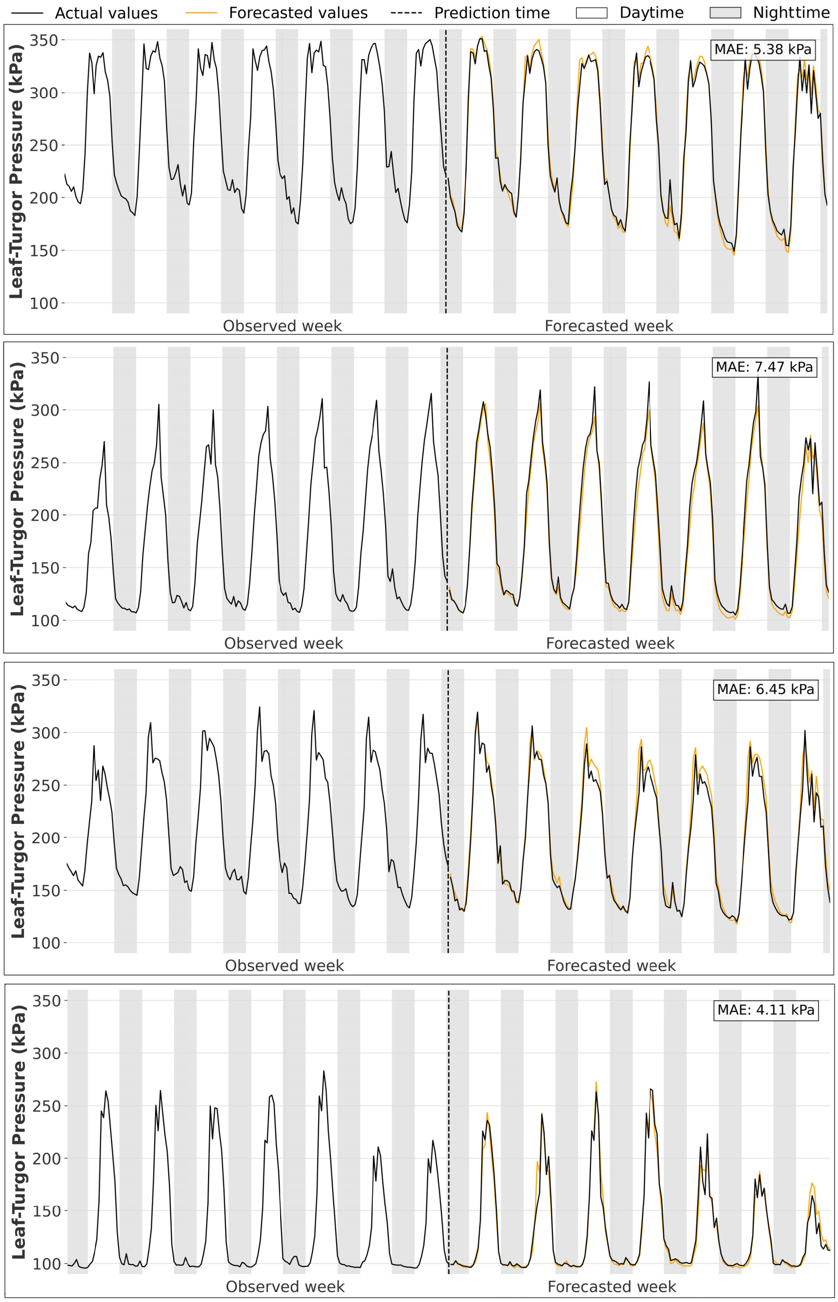


FIGURE 4. Four randomly selected forecasts of P_{leaf} performed by the best model (SVR).

widely used in agriculture to assess crop water stress, among other factors. Higher P_{leaf} values are associated with well-hydrated trees, whereas lower values suggest increased water stress [11], [12], [13]. Despite these variations, the SVR model consistently exhibits high performance in predicting the temporal evolution of P_{leaf}. For instance, in the first forecast, P_{leaf} fluctuates between 150 kPa and 350 kPa, while in the last forecast, values range from 100 kPa to just above 250 kPa. The SVR model successfully captures P_{leaf} variation in both cases, with predicted values closely aligning with the actual observations.

Together with the performance metrics provided in Table 4, the forecasts illustrated in Figure 4 provide strong evidence of the model's robustness and reliability in predicting P_{leaf} dynamics. This further reinforces the suitability of the SVR model for forecasting P_{leaf}, making it a valuable tool for predicting and proactively managing crop conditions in agricultural applications.

B. RESULTS OF T_{LEAF} FORECASTING MODELS

Addressing the T_{leaf} forecasting models, Table 5 presents the performance metrics obtained for each learning model

TABLE 5. Performance metrics obtained for each learning model in forecasting T_{leaf} using the test set.

Algorithm	MAE	MSE	RMSE	R-squared	Hyperparameters
Linear Regression (LR)	0.61	0.70	0.82	0.98	-
Ridge Regression (RR)	0.61	0.70	0.82	0.98	alpha = 0.5
k-Nearest Neighbors Regression (KNNR)	0.92	1.73	1.28	0.96	n_neighbors = 27, metric = "manhattan", weights = "uniform"
Support Vector Regression (SVR)	**0.36**	**0.41**	**0.59**	**0.99**	**C = 1, epsilon = 0.00001**
CatBoost Regression (CatBoost)	0.65	0.93	0.93	0.98	iterations = 1000, learning_rate = 0.05, depth = 4
LightGBM Regression (LGBM)	0.65	0.91	0.92	0.98	num_leaves = 50, learning_rate = 0.05, n_estimators = 600

in forecasting T_{leaf} using the test set. As observed in the forecasting of P_{leaf} (Table 4), the predictive performance of the models varies significantly, with SVR once again achieving the best results. SVR records the lowest MAE (0.36), MSE (0.41), and RMSE (0.59), alongside the highest R-squared value (0.99). This R-squared value indicates that 99% of the T_{leaf} variance in the forecasted week is explained by the model, showing its strong ability to capture the underlying temporal patterns of T_{leaf}.

Conversely, KNNR exhibits the weakest performance, yielding the highest errors (MAE = 0.92, RMSE = 1.28) and the lowest R-squared value (0.96), similar to its relatively poor performance in forecasting P_{leaf}. This suggests that KNNR struggles to model the complex physiological processes governing both T_{leaf} and P_{leaf}. LR and RR show nearly identical performance, both achieving an MAE of 0.61 and an R-squared value of 0.98, indicating reasonable predictive capacity but with lower precision compared to SVR.

The gradient boosting models, CatBoost and LGBM, perform moderately well, each achieving an MAE of 0.65 and an R-squared value of 0.98. Although these models perform slightly worse than SVR, their strong predictive capabilities further highlight their robustness as alternative forecasting approaches. These results align with those obtained in P_{leaf} forecasting, where they also exhibited moderate performance.

Similar to P_{leaf}, in order to further evaluate the predictive ability of the best-performing model for T_{leaf} (i.e., SVR), Figure 5 presents four randomly selected forecasts. As can be seen in Figure 5, the predicted values closely align with the actual measurements for the forecasted week, with the model effectively capturing the fluctuations of T_{leaf}. The inclusion of the MAE for each forecast proves that they are representative of the average model performance. In this context, the strong predictive capability of the SVR model is particularly noteworthy, as T_{leaf} differs from ambient temperature, being influenced not only by weather conditions but also by internal plant processes. Notably, T_{leaf} plays a crucial role in assessing crop water stress and affects key factors such as plant growth, evapotranspiration, or disease outbreaks [3]. Together with the performance metrics presented in Table 5, these results reinforce the model's reliability and highlight its potential for improving agricultural decision-making through T_{leaf} forecasting.

V. DIGITAL TWIN SOFTWARE ARCHITECTURE FOR THE APPLICATION OF THE FORECASTING MODELS

This section proposes a complete digital twin software architecture integrating the machine learning models trained in this study for the forecasting of P_{leaf} and T_{leaf}. First, the motivation for this system is discussed. Next, the essential components and elements that constitute a digital twin system are defined from a domain-agnostic perspective. Building on this foundation, the proposed digital twin software architecture is presented in detail.

A. MOTIVATION

Michael Grieves introduced the "digital twin" concept in 2002, describing it as the integration of a physical system in the real world, a virtual representation in a digital space, and a bidirectional data link connecting the two [52]. Since then, digital twins have emerged as a transformative technology in a variety of industries, enabling real-time monitoring, simulation, optimization, and informed decision-making.

In the agricultural sector, digital twin technology holds considerable promise, particularly in advancing the United Nations Sustainable Development Goals (UN SDGs). For instance, digital twins can contribute to SDG 2 (Zero Hunger) [53] by optimizing resource allocation to improve crop yields, SDG 6 (Clean Water and Sanitation) [54] and SDG 12 (Responsible Consumption and Production) [55] by improving irrigation efficiency and reducing water waste, and SDG 13 (Climate Action) [56] by supporting more sustainable agricultural practices, among others.

However, despite the growing adoption of digital twin technology in sectors such as manufacturing, energy production, and construction, its application in agriculture remains limited and is still in its infancy [28], [29]. A recent cross-domain mapping study revealed a significant disparity in the field of application, with over 70% of digital twin research focusing on manufacturing, and less than 1% on agriculture [29]. In addition, within the agricultural sector, most existing digital twin studies focus on physical assets

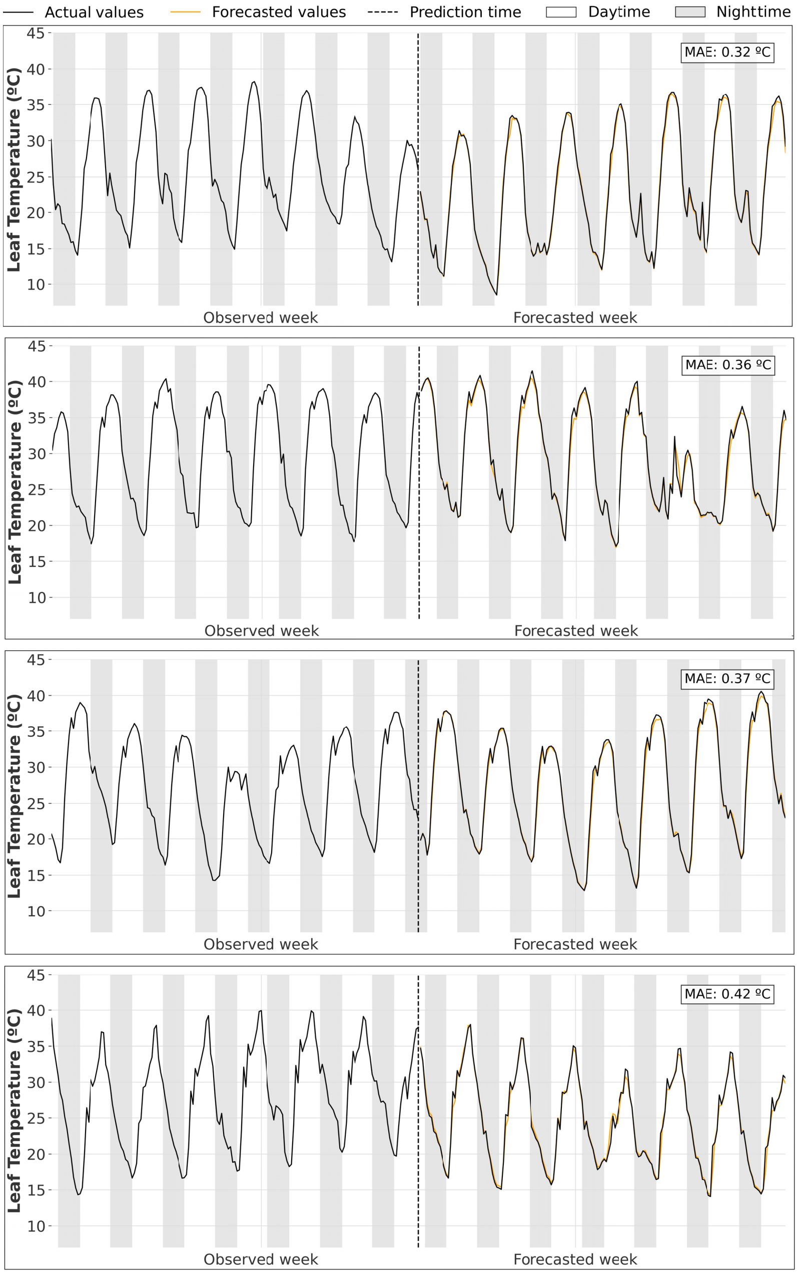


FIGURE 5. Four randomly selected forecasts of T_{leaf} performed by the best model (SVR).

such as greenhouses, with only 9% of studies focusing on plants themselves [30].

Moreover, digital twins can be classified based on their capabilities, including monitoring, predictive, prescriptive, autonomous, and recollection digital twins [57]. Among these, predictive digital twins are of particular relevance to agriculture. These systems leverage (near) real-time data and advanced analytics, such as machine learning models, to forecast the future conditions of their physical counterpart [57]. In agricultural applications, predictive digital twins offer important benefits in anticipating variables such as crop yields, disease outbreaks, and the impact of weather conditions on plant status [30]. However, despite their potential, most existing agricultural digital twin systems are designed for monitoring purposes and only 19% incorporate predictive capabilities [30], which are crucial for long-term planning and decision-making. Therefore, there is a clear need to promote the adoption and advancement of digital twin technology in agriculture, particularly in the context of predictive capabilities.

In the following subsections, the five-dimensional model of digital twins is first presented, defining the essential elements, components, and relationships that a digital twin system should have. Building on this conceptual framework,

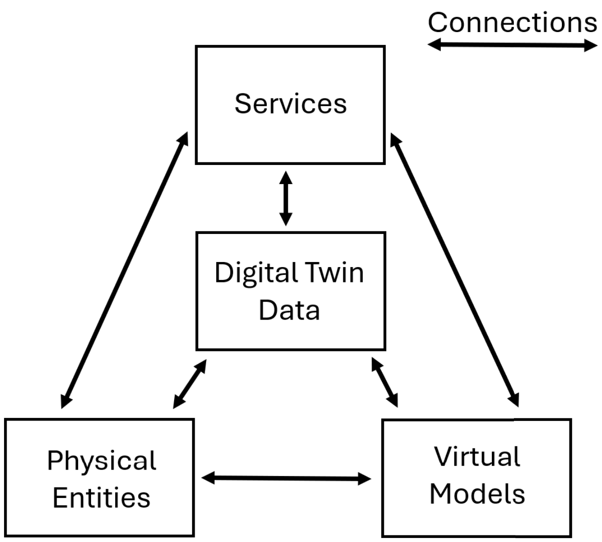


FIGURE 6. Five-dimensional model of digital twins. Adapted from Tao et al. [59].

a comprehensive digital twin software architecture is then proposed. This digital twin architecture integrates the machine learning models trained in this work for forecasting key physiological parameters of crop leaves (i.e., P_{leaf} and T_{leaf}). By exploiting these forecasting models, the proposed system enables the prediction of future crop conditions. Thus, this study not only investigates the development of forecasting models for critical agricultural parameters (i.e., P_{leaf} and T_{leaf}) but also explores how these models can be effectively incorporated into digital twin environments to enable predictive capabilities.

B. THE FIVE DIMENSIONS OF DIGITAL TWINS

Although there is no complete agreement in the literature on what represents a digital twin system [29], [58], certain aspects consistently appear in the different studies. In this sense, the five-dimensional model of digital twins [59] emerged as a structured framework for understanding the basic components and relationships that a digital twin system should have. This five-dimensional model is an extension of the three-dimensional model initially proposed [60], and has been studied and adopted by numerous authors [61], [62], [63], [64]. For these reasons, it is used as a reference in this work.

As illustrated in Figure 6, the five-dimensional model of digital twins includes the *Physical Entities Dimension*, *Digital Twin Data Dimension*, *Virtual Models Dimension*, *Services Dimension*, and *Connections Dimension*. Each of them is presented below from a domain-agnostic perspective.

- *Physical Entities Dimension:* This dimension represents the physical object, process, or system that the digital twin replicates, encompassing industrial machinery, buildings, and even living organisms. Through integration with an IoT system, comprising sensors and actuators, the digital twin enables a continuous bidirectional exchange of data between the physical entity and its virtual counterpart, ensuring real-time synchronization and interaction.
- *Digital Twin Data Dimension:* This dimension focuses on the ingestion, storage, and management of data generated by the digital twin and other external sources. This dimension is essential to ensure the integrity, accuracy, and consistency of the data throughout its lifecycle.
- *Virtual Models Dimension:* This dimension encompasses simulation models that enable the behavior and dynamics of the physical system to be replicated in the virtual world. It includes expert-defined models and data-driven models. These virtual models enable digital twins to analyze system performance, optimize processes, and predict potential issues, thereby enhancing decision-making and improving operational efficiency.
- *Services Dimension:* This dimension encompasses all the capabilities offered by the digital twin. In addition, it includes user-friendly tools and graphical interfaces, such as control panels or cockpits, designed to enable end users to leverage these capabilities.
- *Connections Dimension:* This dimension facilitates communication and data exchange across all components of the digital twin ecosystem. It includes IoT-enabled data acquisition, cloud-based integration, and standardized communication protocols, such as MQTT, HTTP, and RESTful APIs.

Therefore, designing a digital twin integrating the P_{leaf} and T_{leaf} forecasting models trained in this work requires careful consideration of these five dimensions. In the next section, the proposed software architecture is presented.

C. PROPOSED DIGITAL TWIN SOFTWARE ARCHITECTURE

The envisioned digital twin software architecture (see Figure 7) integrates the P_{leaf} and T_{leaf} forecasting models, enabling not only real-time monitoring but also predictive analytics for a Japanese plum orchard (i.e., physical counterpart). This architecture is designed around the five-dimensional model of digital twins, as outlined in Section V-B. How each of these dimensions has been addressed is discussed below.

1) PHYSICAL ENTITIES DIMENSION

The physical entities in the proposed digital twin architecture consist of the Japanese plum orchard and its associated mist layer IoT devices (see Figure 7). These mist layer IoT devices include P_{leaf} and T_{leaf} sensors that provide continuous, (near) real-time data about the physiological state of the trees. Additionally, a weather station located in the orchard gathers environmental data such as ambient temperature, humidity, and solar radiation. These sensors collect valuable information on the status of the trees and

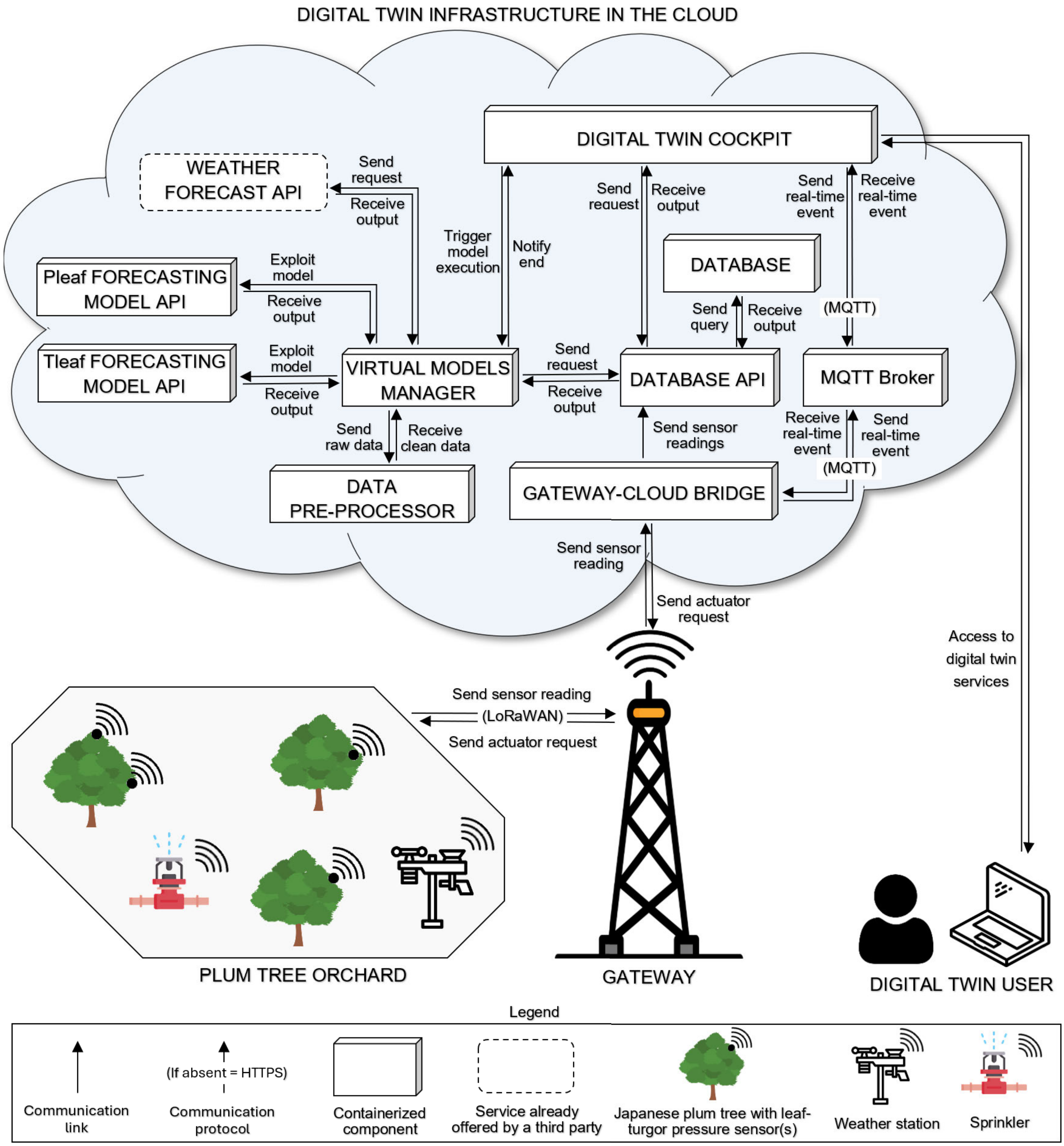


FIGURE 7. **Proposed digital twin software architecture integrating the P_{leaf} and T_{leaf} forecasting models. Icons from Flaticon.com have been used in this illustration.**

the surrounding environment, forming the foundation for the digital twin. The measured data are transmitted to a nearby gateway (through the LoRaWAN protocol), which aggregates the sensor measurements and prepares them for transmission to the cloud, where the rest of the digital twin infrastructure resides.

In addition, the mist layer IoT devices include actuators such as water sprinklers. These actuators are triggered when specific conditions are met, such as P_{leaf} and T_{leaf} values indicating severe water stress. Moreover, users are able to manually activate these actuators through the digital twin cockpit. Thus, the proposed digital twin system supports a bidirectional data flow, where sensor data are transmitted to the virtual replica, and control commands are sent back to the physical counterpart, adjusting orchard conditions as needed.

2) DIGITAL TWIN DATA DIMENSION

As depicted in Figure 7, the collected data from the gateway are first transmitted to a gateway-cloud bridge component, which serves as the intermediary between the edge devices (i.e., the gateway) and the rest of the digital twin infrastructure hosted in the cloud.

The gateway-cloud bridge is primarily responsible for ensuring that the incoming sensor data are appropriately stored and made accessible for subsequent processing within the digital twin system. Upon receiving the data, the edge-cloud bridge stores it in a persistent database, thereby ensuring that the data generated from the physical counterpart (i.e., the orchard) are preserved and available for future use.

In addition to data storage, the gateway-cloud bridge also facilitates real-time event communication between the gateway and the digital twin cockpit. There are two primary types of real-time events: (1) actuator commands and (2) incoming sensor measurements. Specifically, this component handles the reception of actuator commands from the digital twin cockpit via an MQTT broker and forwards these requests to the gateway. Additionally, when new sensor measurements are received, the system promptly sends this data to the digital twin cockpit, ensuring that the interface reflects the most up-to-date information in (near) real-time.

Furthermore, this dimension also incorporates external data sources maintained by third parties. For instance, the Accuweather API [38], to retrieve weather forecasts for the upcoming days. These forecasts include a range of weather parameters, such as temperature, humidity, solar radiation, and other relevant environmental factors.

In this dimension, data pre-processing functions are also handled, which are essential to transform raw data into a suitable format for further processes, particularly for use in machine learning models. A dedicated microservice, implemented as a REST API, is responsible for performing tasks such as data cleaning, normalization, and filtering of the raw sensor inputs. When a request is made, this microservice receives the raw data in JSON format, along with instructions specifying the pre-processing techniques to be applied. These techniques, which might include min-max normalization or time-based sampling, are crucial for ensuring that the data are accurate, consistent, and structured appropriately for use in machine learning models and other services.

3) VIRTUAL MODELS DIMENSION

This dimension includes the machine learning models developed in this work to forecast P_{leaf} and T_{leaf} in Japanese plum trees. These models enable the digital twin to reproduce not only the current state of the orchard (which is achieved by collecting real-time data from sensors) but also the behavior and dynamics of the physical counterpart, by capturing the complex interactions between plant physiological parameters and environmental factors.

In this context, once the P_{leaf} and T_{leaf} forecasting models have been trained, they are serialized. In this process, the trained models, including all learned parameters, are converted into a standardized format (e.g., a.pkl file [65]). It should be noted that serialization allows the models to be stored in a portable form, facilitating their transfer and reuse across different systems without the need for retraining. After serialization, the models are made accessible via REST APIs, allowing other components of the digital twin to interact with them. This design enables the models to be executed on demand, allowing the system to request predictions in real-time based on the most recent sensor data and weather forecasts.

To achieve this, the virtual models manager component coordinates the execution of the machine learning models as needed. This component ensures the operation of the models by managing their execution within a well-defined pipeline, guaranteeing that all necessary steps are followed. The structure of this pipeline is outlined below.

1) First, the virtual models manager component queries the persistent database as well as external data sources through their APIs to retrieve the necessary past data and weather forecast information.
2) Next, all retrieved data are aggregated, and the aggregated data are then passed to the data pre-processor component, which performs the necessary transformations to prepare it for machine learning models. This step includes data subsampling and normalization.
3) Once the data are pre-processed, it is forwarded to the appropriate learning model, either for forecasting P_{leaf} or T_{leaf}. The model then executes inference to generate predictions based on the input data.
4) Next, the resulting predictions are stored back in the persistent database, ensuring that the output is available for future use, analysis, or integration with other system components.
5) Finally, after the predictions are generated and stored, a notification is sent to the digital twin cockpit to update the interface with the latest information, providing real-time insights for the user.

Therefore, the components of the *Virtual Models Dimension* enable the digital twin to replicate the behavior of the orchard in the virtual world by using the machine learning models developed in this work for forecasting P_{leaf} and T_{leaf}. Through serialization and REST APIs, these models are stored and made accessible for on-demand execution. Then, the virtual models manager orchestrates the execution pipeline, ensuring that the models are exploited when needed, serving as the foundation for predictive services within the digital twin.

4) SERVICES DIMENSION

The *Services Dimension* encompasses all the capabilities of the digital twin offered for end users, such as orchard managers, agronomists, and farmers. Through a web-based cockpit, users interact with the digital twin system to monitor the current status of the orchard, predict its future conditions, and make informed decisions. Additionally, the cockpit enables users to send actuator commands, such as opening irrigation sprinklers, to control and influence the conditions of the physical counterpart in a bidirectional manner.

5) CONNECTIONS DIMENSION

The *Connections Dimension* enables communication between the various components of the digital twin system. This dimension is transversal, extending across the entire digital twin infrastructure, and encompasses all communication mechanisms and protocols, including the database API, to facilitate access to persistent data, and the MQTT broker, to handle real-time events.

6) DEPLOYMENT

The proposed digital twin software architecture, as illustrated in Figure 7, has been designed for cloud deployment using Docker containers [66]. This architecture allows all system components, including data pre-processing services, machine learning models, databases, user interfaces, and the virtual models manager, among others, to be containerized. Containerization ensures that each component is isolated and packaged with all necessary dependencies, enabling consistent and reproducible deployments across various environments. This approach provides several advantages, including improved scalability, easier maintenance, and seamless integration with cloud services.

In this context, Kubernetes [67] would be employed for container orchestration, automating the deployment, scaling, and management of the system's components. This allows the system to efficiently handle dynamic and growing computational demands, scaling individual components based on workload requirements.

To sum up, the proposed digital twin software architecture offers a robust framework for integrating the P_{leaf} and T_{leaf} forecasting models. Based on the five-dimensional model presented in Section V-B, it offers a comprehensive approach that aligns with widely accepted digital twin principles and specifications. Furthermore, the containerized design of the system, with all components deployed as Docker containers and orchestrated via Kubernetes, enhances scalability, fault tolerance, and maintainability. Ultimately, this architecture illustrates how the forecasting models trained in this work can be effectively integrated within digital twin environments to enable advanced predictive capabilities, addressing a critical gap in the current research landscape and advancing precision farming technologies.

VI. DISCUSSION

This section discusses the research questions that initially guided the present study. The main objective of this analysis is to highlight the contributions and achievements derived from this work, as well as to elaborate on the implications of its findings in the field of agriculture. In addition, some limitations are also addressed.

Regarding RQ1: *to what extent is it possible to forecast leaf physiological parameters (i.e., T_{leaf} and P_{leaf}) in Japanese plum trees using machine learning and IoT technologies?* The leaf physiological parameters under study are largely influenced by weather conditions and exhibit a specific behavior depending on the health status, water stress, genetics, and other characteristics of the plant. However, using technologies such as IoT and machine learning, it is possible to predict, to a large extent, the future variation of the leaf physiological parameters studied in Japanese plum trees. Specifically, using past measurements of the target leaf variable at the prediction time along with past and future weather conditions, the variation in the future week of the target variable can be forecasted with a 0.96 R-squared in the case of P_{leaf}, and 0.99 in the case of T_{leaf}. Note that R-squared is the proportion of variation of the target variable explained by the predictor variables.

As for RQ2: *what is the performance of the learning models in forecasting T_{leaf} and P_{leaf} in Japanese plum trees with a seven-day forecast horizon?* and RQ3: *which machine learning algorithms are the most effective for training T_{leaf} and P_{leaf} forecasting models?* Using the best-performing model, it is possible to predict the values of the future week of P_{leaf} and T_{leaf} with R-squared values of 0.96 and 0.99, respectively. In both cases, the best-performing model was trained by means of the Support Vector Regression algorithm, which has proven to be the most effective for this purpose among those considered in this work.

Based on the results obtained for RQ1, RQ2, and RQ3, the integration of machine learning and IoT technologies holds great potential for fostering proactive management in agriculture. T_{leaf} and P_{leaf} are key indicators of crop status, and the proposed approach has proven to be highly effective for forecasting these physiological parameters with a one-week time horizon. These predictive capabilities would allow farmers to prematurely identify potential problems, such as water stress, before they affect crop health or quality, allowing a shift from reactive to proactive management. Ultimately, this can lead to improved crop yields, reduced environmental impact, and more sustainable farming practices.

With respect to RQ4: *How could the T_{leaf} and P_{leaf} forecasting models be integrated into digital twin systems to support predictive capabilities?* To answer this question, a comprehensive digital twin software architecture integrating the T_{leaf} and P_{leaf} forecasting models has been proposed. This architecture is based on the five-dimensional model of digital twins, which provides a structured framework for defining the essential components of a digital twin system. The five dimensions include: the *Physical Entities*

Dimension, the *Digital Twin Data Dimension*, the *Virtual Models Dimension*, the *Services Dimension*, and the *Connections Dimension*.

Within this architecture, once the forecasting models are trained, they are serialized and exposed through REST APIs to enable on-demand execution. When required, the virtual models manager component coordinates the execution of these models through a structured pipeline. This process involves querying relevant data sources, aggregating the data, sending it for pre-processing, executing the appropriate machine learning model, and persistently storing the output. By following this approach, the digital twin system can apply the forecasting models to deliver advanced predictive capabilities that support proactive, data-driven decision-making in agriculture.

Therefore, this work also holds significant potential for advancing digital twin technology in agriculture, particularly predictive digital twins. Moreover, it should also be noted that one of the key steps in the development of a digital twin is the creation of accurate virtual models that faithfully reproduce the behavior and dynamics of the physical counterpart in the virtual world (see *Virtual Models Dimension* in Section V-B). In this sense, while this paper focuses on forecasting P_{leaf} and T_{leaf} in Japanese plum trees, this study can also help other researchers and practitioners to obtain data-driven models for forecasting other agricultural variables.

Furthermore, considering that sensors deployed on crop leaves tend to fail as they are exposed to a hostile environment (weather conditions, animal activities, growth of the leaves themselves, etc.), this novel approach to forecasting leaf physiological parameters could improve the robustness and reduce the maintenance costs of leaf-attached sensors in agriculture. First, the learning models could make it possible to recover missing and faulty measurements by predicting them. Second, this approach could also make it possible to detect faults and anomalies in the operation of the sensors (i.e., identifying when measured values differ substantially from those expected).

VII. CONCLUSION AND FUTURE WORK

Leaf-Turgor Pressure (P_{leaf}) and Leaf Temperature (T_{leaf}) are physiological parameters widely used in agriculture to assess crop condition and guide farm management. These variables are typically measured by Internet of Things (IoT) systems involving sensors attached to crop leaves. However, although these sensors provide (near) real-time information on the physiological state of plants, they do not enable proactive farm management, which is increasingly necessary in the current context of accelerating climate change and uncertainty.

In this paper, the combination of machine learning and IoT technologies is proposed to enable not only real-time monitoring but also forecasting of leaf physiological parameters (i.e., P_{leaf} and T_{leaf}). For this purpose, an experimental field of Japanese plum trees of *Angeleno*, *Talete*, and *Fortune* cultivars has been monitored for five months; in addition, meteorological conditions were recorded. Using the data gathered, machine learning algorithms have been applied to train P_{leaf} and T_{leaf} forecasting models. Results report Support Vector Machines as the most effective algorithm, making it possible to predict P_{leaf} and T_{leaf} in Japanese plum trees one week in advance with R-squared values of 0.96 and 0.99, respectively.

Thus, this work represents a significant breakthrough in anticipating the future state of crops, paving the way towards more proactive and resilient farming practices, as illustrated by the proposed digital twin, which integrates the P_{leaf} and T_{leaf} forecasting models to provide advanced predictive capabilities.

Finally, considering the contributions and results of this study, three avenues for future research are presented. First, the proposed approach could be applied to enable the forecasting of other agricultural parameters, allowing a more complete understanding of the future state of crops. Second, the forecasting models could be used to detect erroneous measurements from sensors installed on crop leaves by identifying discrepancies between measured and expected values. In addition, these models could be used to recover erroneous or missing readings by forecasting them. In this sense, it should be recalled that sensors in agriculture are exposed to a hostile environment (e.g., weather conditions and animal activities), and exploring strategies to improve their resilience to errors is an open area of research.

ACKNOWLEDGMENT

Proyecto TED2021-129194B-I00 financiado por:

Expediente GR24099

Expediente 59391100602202300000186; PD23034

REFERENCES

[1] International Labour Organization, Geneva, Switzerland. (2021). *World Employment and Social Outlook 2021: Trends 2021*. [Online]. Available: https://www.ilo.org/sites/default/files/wcmsp5/groups/public/%40dgreports/%40dcomm/%40publ/documents/publication/wcms_795453.pdf

[2] *The State of Food and Agriculture 2020. Overcoming Water Challenges in Agriculture*, FAO, Rome, Italy, 2020.

[3] Y. Lu, W. Wang, X. Zhang, and W. Zheng, "A review on leaf temperature sensor: Measurement methods and application," in *Proc. 9th IFIP WG 5.14 Int. Conf. Comput. Comput. Technol. Agricult.*, Beijing, China. Cham, Switzerland: Springer, Sep. 2016, pp. 216–230.

[4] W. Fricke, "Turgor pressure," in *Encyclopedia of Life Sciences*. Wiley, pp. 1–6, 2017, doi: 10.1002/9780470015902.a0001687.pub2.

[5] P. Ache, H. Bauer, H. Kollist, K. A. S. Al-Rasheid, S. Lautner, W. Hartung, and R. Hedrich, "Stomatal action directly feeds back on leaf turgor: New insights into the regulation of the plant water status from non-invasive pressure probe measurements," *Plant J.*, vol. 62, no. 6, pp. 1072–1082, Apr. 2010.

[6] M. Ouhami, A. Hafiane, Y. Es-Saady, M. El Hajji, and R. Canals, "Computer vision, IoT and data fusion for crop disease detection using machine learning: A survey and ongoing research," *Remote Sens.*, vol. 13, no. 13, p. 2486, Jun. 2021.

[7] J. Peñuelas, R. Savé, O. Marfà, and L. Serrano, "Remotely measured canopy temperature of greenhouse strawberries as indicator of water status and yield under mild and very mild water stress conditions," *Agricult. Forest Meteorol.*, vol. 58, nos. 1–2, pp. 63–77, Mar. 1992.
[8] S. Gutiérrez, M. P. Diago, J. Fernández-Novales, and J. Tardaguila, "Vineyard water status assessment using on-the-go thermal imaging and machine learning," *PLoS ONE*, vol. 13, no. 2, Feb. 2018, Art. no. e0192037.
[9] J. A. Barriga, F. Blanco-Cipollone, E. Trigo-Córdoba, I. García-Tejero, and P. J. Clemente, "Crop-water assessment in citrus (Citrus sinensis L.) based on continuous measurements of leaf-turgor pressure using machine learning and IoT," *Expert Syst. Appl.*, vol. 209, Dec. 2022, Art. no. 118255.
[10] H. Chehab, M. Tekaya, B. Mechri, A. Jemai, M. Guiaa, Z. Mahjoub, D. Boujnah, S. Laamari, B. Chihaoui, H. Zakhama, M. Hammami, and T. del Giudice, "Effect of the super absorbent polymer Stockosorb on leaf turgor pressure, tree performance and oil quality of olive trees cv. Chemlali grown under field conditions in an arid region of Tunisia," *Agricult. Water Manage.*, vol. 192, pp. 221–231, Oct. 2017.
[11] M. A. Martínez-Gimeno, M. Castiella, S. Rüger, D. S. Intrigliolo, and C. Ballester, "Evaluating the usefulness of continuous leaf turgor pressure measurements for the assessment of persimmon tree water status," *Irrigation Sci.*, vol. 35, no. 2, pp. 159–167, Mar. 2017.
[12] L. B. Riboldi, R. F. Oliveira, and L. R. Angelocci, "Leaf turgor pressure in maize plants under water stress," *Austral. J. Crop Sci.*, vol. 10, no. 6, pp. 878–886, Jun. 2016.
[13] C. M. Padilla-Díaz, C. M. Rodriguez-Dominguez, V. Hernandez-Santana, A. Perez-Martin, and J. E. Fernández, "Scheduling regulated deficit irrigation in a hedgerow olive orchard from leaf turgor pressure related measurements," *Agricult. Water Manage.*, vol. 164, pp. 28–37, Jan. 2016.
[14] C. M. Rodriguez-Dominguez, W. Ehrenberger, C. Sann, S. Rüger, V. Sukhorukov, M. J. Martín-Palomo, A. Diaz-Espejo, M. V. Cuevas, J. M. Torres-Ruiz, A. Perez-Martin, U. Zimmermann, and J. E. Fernández, "Concomitant measurements of stem sap flow and leaf turgor pressure in olive trees using the leaf patch clamp pressure probe," *Agricult. Water Manage.*, vol. 114, pp. 50–58, Nov. 2012.
[15] R. D. M. Fernandes, M. V. Cuevas, V. Hernandez-Santana, C. M. Rodriguez-Dominguez, C. M. Padilla-Díaz, and J. E. Fernández, "Classification models for automatic identification of daily states from leaf turgor related measurements in olive," *Comput. Electron. Agricult.*, vol. 142, pp. 181–189, Nov. 2017.
[16] R. Chen, Q. Xu, J. Wan, N. Jiang, J. Ma, and L. Zheng, "The daily minimum leaf turgor pressure can represent the water status of apple trees under drip irrigation," *Frontiers Plant Sci.*, vol. 15, Dec. 2024, Art. no. 1516824.
[17] G. Camoglu, K. Demirel, F. Kahriman, A. Akcal, H. Nar, A. Boran, I. Eroglu, and L. Genc, "Discrimination of water stress in pepper using thermography and leaf turgor pressure probe techniques," *Agricult. Water Manage.*, vol. 254, Aug. 2021, Art. no. 106942.
[18] P. Hewage, M. Trovati, E. Pereira, and A. Behera, "Deep learning-based effective fine-grained weather forecasting model," *Pattern Anal. Appl.*, vol. 24, no. 1, pp. 343–366, Feb. 2021.
[19] M. Panja, T. Chakraborty, S. S. Nadim, I. Ghosh, U. Kumar, and N. Liu, "An ensemble neural network approach to forecast dengue outbreak based on climatic condition," *Chaos, Solitons Fractals*, vol. 167, Feb. 2023, Art. no. 113124.
[20] M. Shokouhifar and M. Ranjbarimesan, "Multivariate time-series blood donation/demand forecasting for resilient supply chain management during COVID-19 pandemic," *Cleaner Logistics Supply Chain*, vol. 5, Dec. 2022, Art. no. 100078.
[21] D. Markovics and M. J. Mayer, "Comparison of machine learning methods for photovoltaic power forecasting based on numerical weather prediction," *Renew. Sustain. Energy Rev.*, vol. 161, Jun. 2022, Art. no. 112364.
[22] O. Mudannayake, D. Rathnayake, J. D. Herath, D. K. Fernando, and M. Fernando, "Exploring machine learning and deep learning approaches for multi-step forecasting in municipal solid waste generation," *IEEE Access*, vol. 10, pp. 122570–122585, 2022.
[23] A. Goap, D. Sharma, A. K. Shukla, and C. R. Krishna, "An IoT based smart irrigation management system using machine learning and open source technologies," *Comput. Electron. Agricult.*, vol. 155, pp. 41–49, Dec. 2018.
[24] D. Paudel, H. Boogaard, A. de Wit, M. van der Velde, M. Claverie, L. Nisini, S. Janssen, S. Osinga, and I. N. Athanasiadis, "Machine learning for regional crop yield forecasting in Europe," *Field Crops Res.*, vol. 276, Feb. 2022, Art. no. 108377.
[25] V. Parkash and S. Singh, "A review on potential plant-based water stress indicators for vegetable crops," *Sustainability*, vol. 12, no. 10, p. 3945, May 2020.
[26] W. Zhao, L. Liu, Q. Shen, J. Yang, X. Han, F. Tian, and J. Wu, "Effects of water stress on photosynthesis, yield, and water use efficiency in winter wheat," *Water*, vol. 12, no. 8, p. 2127, Jul. 2020.
[27] A. Barriga, J. A. Barriga, M. J. Moñino, and P. J. Clemente, "IoT-based expert system for fault detection in Japanese plum leaf-turgor pressure WSN," *Internet Things*, vol. 23, Oct. 2023, Art. no. 100829.
[28] L. Wang, "Digital twins in agriculture: A review of recent progress and open issues," *Electronics*, vol. 13, no. 11, p. 2209, Jun. 2024.
[29] M. Dalibor, N. Jansen, B. Rumpe, D. Schmalzing, L. Wachtmeister, M. Wimmer, and A. Wortmann, "A cross-domain systematic mapping study on software engineering for digital twins," *J. Syst. Softw.*, vol. 193, Nov. 2022, Art. no. 111361.
[30] N. Ariesen-Verschuur, C. Verdouw, and B. Tekinerdogan, "Digital twins in greenhouse horticulture: A review," *Comput. Electron. Agricult.*, vol. 199, Aug. 2022, Art. no. 107183.
[31] M. A. Castán-Lascorz, P. Jiménez-Herrera, A. Troncoso, and G. Asencio-Cortés, "A new hybrid method for predicting univariate and multivariate time series based on pattern forecasting," *Inf. Sci.*, vol. 586, pp. 611–627, Mar. 2022.
[32] T. van Klompenburg, A. Kassahun, and C. Catal, "Crop yield prediction using machine learning: A systematic literature review," *Comput. Electron. Agricult.*, vol. 177, Oct. 2020, Art. no. 105709.
[33] S. F. Crone, S. Lessmann, and R. Stahlbock, "The impact of preprocessing on data mining: An evaluation of classifier sensitivity in direct marketing," *Eur. J. Oper. Res.*, vol. 173, no. 3, pp. 781–800, Sep. 2006.
[34] A. K. Uysal and S. Gunal, "The impact of preprocessing on text classification," *Inf. Process. Manage.*, vol. 50, no. 1, pp. 104–112, Jan. 2014.
[35] Z. Wang, H. Zhu, Z. Dong, X. He, and S. Huang, "Less is better: Unweighted data subsampling via influence function," in *Proc. AAAI Conf. Artif. Intell.*, 2020, vol. 34, no. 4, pp. 6340–6347.
[36] S. G. K. Patro and K. K. Sahu, "Normalization: A preprocessing stage," 2015, *arXiv:1503.06462*.
[37] OpenWeather. (2012). [Online]. Available: https://openweathermap.org/api
[38] AccuWeather. (2025). *Accuweather APIs–Forecast API*. Accessed: Mar. 28, 2025. [Online]. Available: https://developer.accuweather.com/accuweather-forecast-api/apis
[39] Z. Reitermanova et al., "Data splitting," in *Proc. WDS*, vol. 10, 2010, pp. 31–36.
[40] D. C. Montgomery, E. A. Peck, and G. G. Vining, *Introduction to Linear Regression Analysis*. Hoboken, NJ, USA: Wiley, 2021.
[41] A. E. Hoerl and R. W. Kennard, "Ridge regression: Biased estimation for nonorthogonal problems," *Technometrics*, vol. 42, no. 1, p. 80, Feb. 2000.
[42] G. C. McDonald, "Ridge regression," *Wiley Interdiscipl. Rev., Comput. Statist.*, vol. 1, no. 1, pp. 93–100, 2009.
[43] O. Krämer, "K-nearest neighbors," in *Dimensionality Reduction With Unsupervised Nearest Neighbors*. Berlin, Germany: Springer, 2013, pp. 13–23.
[44] M. Awad, R. Khanna, M. Awad, and R. Khanna, "Support vector regression," in *Efficient Learning Machines: Theories, Concepts, and Applications for Engineers and System Designers*. Berkeley, CA, USA: Apress, 2015, pp. 67–80.
[45] A. V. Dorogush, V. Ershov, and A. Gulin, "CatBoost: Gradient boosting with categorical features support," 2018, *arXiv:1810.11363*.
[46] G. Ke, Q. Meng, T. Finley, T. Wang, W. Chen, W. Ma, Q. Ye, and T. Liu, "LightGBM: A highly efficient gradient boosting decision tree," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 3146–3154.
[47] J. Herzen, F. Lässig, S. G. Piazzetta, T. Neuer, L. Tafti, G. Raille, T. V. Pottelbergh, M. Pasieka, A. Skrodzki, N. Huguenin, M. Dumonal, J. Kościsz, D. Bader, F. Gusset, M. Benheddi, C. Williamson, M. Kosinski, M. Petrik, and G. Grosch, "Darts: User-friendly modern machine learning for time series," *J. Mach. Learn. Res.*, vol. 23, no. 1, pp. 5442–5447, 2022.
[48] L. Yang and A. Shami, "On hyperparameter optimization of machine learning algorithms: Theory and practice," *Neurocomputing*, vol. 415, pp. 295–316, Nov. 2020.
[49] P. Liashchynskyi and P. Liashchynskyi, "Grid search, random search, genetic algorithm: A big comparison for NAS," 2019, *arXiv:1912.06059*.
[50] C. Willmott and K. Matsuura, "Advantages of the mean absolute error (MAE) over the root mean square error (RMSE) in assessing average model performance," *Climate Res.*, vol. 30, no. 1, pp. 79–82, 2005.

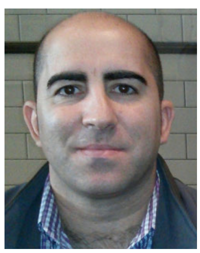
[51] E. Kasuya, "On the use of r and r squared in correlation and regression," Tech. Rep., 2019, doi: 10.1111/1440-1703.1011.
[52] M. Grieves and J. Vickers, "Digital twin: Mitigating unpredictable, undesirable emergent behavior in complex systems," in *Transdisciplinary Perspectives on Complex Systems: New Findings and Approaches*. Cham, Switzerland: Springer, 2017, pp. 85–113.
[53] United Nations. (2015). *Ustainable Development Goal 2: Zero Hunger*. Accessed: Mar. 26, 2025. [Online]. Available: https://sdgs.un.org/goals/goal2
[54] United Nations. (2015). *Sustainable Development Goal 6: Clean Water and Sanitation*. Accessed: Mar. 26, 2025. [Online]. Available: https://sdgs.un.org/goals/goal6
[55] (2015). *Sustainable Development Goal 12: Responsible Consumption and Production*. Sustainable development goal 6: Clean water and sanitation. Accessed: Mar. 26, 2025. [Online]. Available: https://sdgs.un.org/goals/goal12
[56] (2015). *Sustainable Development Goal 13: Climate Action*. Accessed: Mar. 26, 2025. [Online]. Available: https://sdgs.un.org/goals/goal13
[57] C. Verdouw, B. Tekinerdogan, A. Beulens, and S. Wolfert, "Digital twins in smart farming," *Agricult. Syst.*, vol. 189, Apr. 2021, Art. no. 103046.
[58] H. M. Muctadir, D. A. Manrique Negrin, R. Gunasekaran, L. Cleophas, M. van den Brand, and B. R. Haverkort, "Current trends in digital twin development, maintenance, and operation: An interview study," *Softw. Syst. Model.*, vol. 23, no. 5, pp. 1–31, Oct. 2024.
[59] F. Tao, M. Zhang, Y. Liu, and A. Y. C. Nee, "Digital twin driven prognostics and health management for complex equipment," *CIRP Ann.*, vol. 67, no. 1, pp. 169–172, 2018.
[60] M. Grieves, "Digital twin: Manufacturing excellence through virtual factory replication," Tech. Rep., 2014, pp. 1–7. [Online]. Available: https://www.3ds.com/fileadmin/PRODUCTS-SERVICES/DELMIA/PDF/Whitepaper/DELMIA-APRISO-Digital-Twin-Whitepaper.pdf
[61] F. Tao, H. Zhang, A. Liu, and A. Y. C. Nee, "Digital twin in industry: State-of-the-art," *IEEE Trans. Ind. Informat.*, vol. 15, no. 4, pp. 2405–2415, Apr. 2019.
[62] C. Wu, Y. Zhou, M. V. Pereia Pessôa, Q. Peng, and R. Tan, "Conceptual digital twin modeling based on an integrated five-dimensional framework and TRIZ function model," *J. Manuf. Syst.*, vol. 58, pp. 79–93, Jan. 2021.
[63] Q. Qi, F. Tao, T. Hu, N. Anwer, A. Liu, Y. Wei, L. Wang, and A. Y. C. Nee, "Enabling technologies and tools for digital twin," *J. Manuf. Syst.*, vol. 58, pp. 3–21, Jan. 2021.
[64] E. Henrichs, T. Noack, A. M. Pinzon Piedrahita, M. A. Salem, J. Stolz, and C. Krupitzer, "Can a byte improve our bite? An analysis of digital twins in the food industry," *Sensors*, vol. 22, no. 1, p. 115, Dec. 2021.
[65] Python Software Foundation. (2025). *Pickle–Python Object Serialization* Accessed: Mar. 29, 2025. [Online]. Available: https://docs.python.org/3/library/pickle.html
[66] Docker. (2025). *Docker: Accelerated Container Application Development*. Accessed: Mar. 29, 2025. [Online]. Available: https://www.docker.com/
[67] Kubernetes. (2025). *Production-Grade Container Orchestration*. Accessed: Mar. 29, 2025. [Online]. Available: https://kubernetes.io/

ARTURO BARRIGA received the B.Sc. degree from the University of Extremadura, Spain, in 2022, and the M.Sc. degree from the International University of La Rioja, Spain, in 2023. He is currently pursuing the Ph.D. degree with the University of Extremadura. He is a Junior Researcher at the Quercus Software Engineering Group, University of Extremadura. His research interests include model-driven development, digital twins, and machine learning applied to agriculture.

JOSÉ A. BARRIGA received the B.Sc. degree from the University of Extremadura, Spain, in 2018, the M.Sc. degree from the International University of La Rioja, in 2019, and the Ph.D. degree in computer science from the University of Extremadura, in 2024. He is currently a Postdoctoral Researcher at the Quercus Software Engineering Group, University of Extremadura. He has been working for six years in the areas of IoT systems simulation, model-driven-development applied to the IoT, and machine learning applied to agriculture.

ADOLFO LOZANO-TELLO received the B.S. degree in computer science from the University of Granada, Spain, in 1993, and the Ph.D. degree in computer science from the University of Extremadura, Spain, in 2002. He is currently a Teaching/Research Assistant Professor of computer languages and systems at the University of Extremadura. His research interests include reusable software components rating, ontology design, linked data portals, the use of software components and ontologies in applications, the use of ontologies in ambient intelligence, and deep learning applications for satellite crop identification. He has published more than 100 articles on the above issues of software engineering and knowledge engineering. He belongs to several related projects funded by various institutions and companies.

MARÍA J. MOÑINO received the Ph.D. degree in agronomy from the University of Extremadura, in 2011. She is currently a Researcher at the Center for Scientific and Technological Research of Extremadura (CICYTEX). Her research interest includes mitigating the effects that climate change has on semi-arid areas through the implementation of water-saving techniques such as controlled deficit irrigation.



PEDRO J. CLEMENTE received the B.Sc. degree in computer science from the University of Extremadura, Spain, in 1998, and the Ph.D. degree in computer science, in 2007. He is currently an Associate Professor with the Computer Science Department, University of Extremadura. He has published numerous peer-reviewed papers in international journals, workshops, and conferences. His research interests include component-based software development, aspect orientation, service-oriented architectures, business process modeling, and model-driven development. He is involved in several research projects. He has participated in many workshops and conferences as a speaker and a member of the program committees.