Artificial Intelligence-Based Local Weather Forecasting for Agricultural Digital Twins

Miguel Zaragoza-Esquerdo[1], Alberto Ivars-Palomares[1], Irene Eiros-Fonseca[2], Sandra Sendra[1], Jaime Lloret[1]
[1]Instituto de Investigación para la Gestión Integrada de Zonas Costeras, Universitat Politècnica de València
C/Paranimf, 1, 46730 Grao de Gandia, Spain
[2]Universidad CEU San Pablo
C. de Julián Romea, 23, Chamberí, 28003 Madrid
mizaes2@epsg.upv.es, aivapal@epsg.upv.es, i.eiros@usp.ceu.es, sansenco@upv.es, jlloret@dcom.upv.es

***Abstract*—This work presents the development of a local weather forecasting system integrated into an agricultural digital twin, leveraging classical machine learning. Data were collected from ESP32-based weather stations equipped with temperature, relative humidity, and atmospheric pressure sensors. The acquired measurements were processed through a Node.js server and used to train predictive models, including Random Forest, Gradient Boosting, Ridge Regression, Lasso Regression and K-Nearest Neighbors. A sliding window approach was applied to structure the input data for short-term forecasting. Experimental results show that Gradient Boosting achieved the best performance among classical methods for atmospheric pressure but exhibited overfitting for temperature and humidity. These findings highlight the potential of Artificial Intelligence (AI)-powered digital twins to enhance precision agriculture by providing accurate, localized, and up-to-date weather forecasts.**

Keywords-Digital twin; local weather forecasting; machine learning; deep learning; agriculture; Gradient Boosting.

I. INTRODUCTION

In recent decades, technological advancements have driven the development of digital twins, virtual replicas of real-world physical systems that enable real-time monitoring, simulation, and control. In the agricultural sector, these digital twins integrate data from sensors, weather stations, and Internet of Things (IoT) devices to remotely model and optimize complex processes, facilitating the management of resources such as water, fertilizers, and pesticides [1].

The reliability of these models largely depends on the underlying sensing infrastructure. Well-designed Wireless Sensor Network (WSN) architectures, such as those proposed by Lloret et al. [2], and Hussein et al. [3] have proven essential for ensuring coverage, scalability, and efficiency in data collection. Such infrastructures enable parallel and organized communication between multiple nodes, optimizing network topology and reducing latency in transmitting critical data.

By replicating plants and cultivation environments, digital twins offer farmers decision support systems that reduce resource consumption and improve productivity [4]. These virtual representations facilitate scenario evaluation and predictive analysis without extensive physical trials, accelerating the digital transformation of the agricultural sector [5]. Incorporating artificial intelligence and machine learning expands the capabilities of these models, allowing, for example, adaptive irrigation scheduling based on real-time soil moisture data and weather forecasts, significantly reducing water waste [6].

Artificial intelligence techniques have emerged as a key component in high-precision weather forecasting, leveraging convolutional and recurrent neural networks to model complex atmospheric phenomena [7]. These approaches have been shown to improve the estimation of precipitation and extreme temperatures, enabling digital twins to anticipate adverse conditions and proactively adapt their crop management strategies [8].

Moreover, digital twins support sustainable agricultural practices through continuous environmental monitoring and adaptive management strategies [9]. The combination of cloud computing technologies and edge devices increases data processing capacity and allows faster, more precise responses to changes in the field [10].

This work presents the design and implementation of a local weather forecasting system integrated into an agricultural digital twin. Unlike previous studies, our proposal combines IoT-based sensing infrastructure with classical machine learning models to generate short-term, high-resolution forecasts directly tailored to the conditions of a specific agricultural plot. The system demonstrates the feasibility of deploying low-cost weather stations with real-time data processing and highlights the comparative advantages and limitations of different predictive approaches. This contribution provides a practical framework for enhancing decision-making in precision agriculture through accurate, localized, and continuously updated forecasts for digitals twins data input.

The remainder of this article is structured as follows: Section II reviews related works on digital twins, artificial intelligence, and weather forecasting in agricultural contexts. Section III details the architecture and operation of the proposed system, including data acquisition, preprocessing, and model training. Section IV presents and discusses the experimental results, covering machine learning approaches. Finally, Section V summarizes the conclusions and outlines potential future research directions.

II. Related Work

Several studies have reviewed the adoption of digital twins in agriculture, focusing on controlled environments such as greenhouses. A systematic review highlights how these models improve horticultural productivity and sustainability, emphasizing their role in microclimate control, crop growth monitoring, and resource use efficiency [11].

Integrating sensors to measure humidity, temperature, CO_2 levels, and light, along with IoT platforms, increases the accuracy of growth simulations and optimizes climate control in greenhouses [12]. As documented by Bri et al. [13], real deployments of wireless sensor networks demonstrate the feasibility and challenges associated with large-scale agricultural monitoring, including node resilience in adverse environments and energy management to ensure continuity of real-time measurements.

Virtual and augmented reality technologies have strengthened digital twin platforms by enabling immersive interactions with virtual crop models. Examples like the “Virtual Breeding Nursery” allow farmers to explore virtual plots, manipulate environmental variables, and simulate stress or pest infestation scenarios [14]. These intuitive data visualization interfaces combine sensor information with 3D models of plants and structures, supporting more informed decision-making [15].

In open-field contexts, pilot projects demonstrate that platforms equipped with IoT sensor networks—streaming real-time soil moisture, nutrient content, and weather data—can dynamically adapt management practices, optimizing fertilizer use and pest control while minimizing environmental impact [16]. Cooperative group-based solutions, such as those presented by García et al. [17], reduce energy consumption and improve communication efficiency in WSNs, increasing the viability of these platforms in rural areas with limited infrastructure.

Advanced Deep Learning techniques have been successfully applied to environmental monitoring, enabling the early detection of anomalous patterns in air quality, humidity, and temperature [18]. This automated analysis facilitates the integration of predictive systems within digital twins, enhancing responsiveness to emerging weather events or pest outbreaks. Nevertheless, challenges remain, such as data interoperability, real-time synchronization, and affordability for smallholder farmers, which currently limit widespread adoption [19].

III. System Proposal

This section presents a proposed system for implementing local weather forecasting based on artificial intelligence within a digital twin environment.

A. System Description

The proposed system is based on weather stations developed on the ESP32 platform, as illustrated in Figure 1. These stations are equipped with sensors capable of measuring key meteorological variables such as air temperature, relative humidity, and atmospheric pressure.

The data captured by the sensors is transmitted and managed through a server implemented with Node.js, which allows for their storage, processing, and subsequent use for modeling. The locally obtained meteorological variables constitute the inputs to the prediction model based on a machine learning models.

The management of weather forecast requests and the visualization of real-time data and model results is carried out through a user interface developed as part of an application for monitoring agricultural fields. This application serves a dual purpose: to facilitate interaction with the system and as a key component in creating a digital twin of the agricultural environment.

Figure 1. Photograph of the weather station.

B. Operating Algorithm

The weather station's operating algorithm operates in a loop with an execution frequency of once per hour. After this interval, the node automatically connects to the wireless network, establishes communication with the sensors using the Inter-Integrated Circuit (I2C) or Universal Asynchronous Receiver-Transmitter (UART) protocols, and measures the meteorological variables.

Once the data is obtained, the system attempts to connect with the server. If the connection is successful, the data is sent, and the node again enters a standby state until the next iteration of the cycle.

Regarding the operation of the digital twin, the interaction begins when the user clicks the forecast button on the interface associated with a specific weather station. This action activates the weather prediction model, which generates a forecast for the next 24 hours.

Additionally, the data collected daily by the weather stations is used to update the model. In this way, a new model version is trained daily, ensuring that predictions are based on the most recent local weather records, thereby increasing the system's accuracy and adaptability to environmental conditions. This data is used for input for digital twins.

C. Dataset Description

The dataset used in this work corresponds to a time series of local meteorological observations recorded from February 1, 2025 to August 1, 2025. During this period, atmospheric conditions were assessed intermittently over a specific agricultural plot. The dataset is composed of the following variables:

- Air temperature (°C)
- Relative humidity (%)
- Atmospheric pressure (hPa)
- Date
- Time

Each observation is associated with a timestamp (date and time) indicating the exact moment the measurement was taken. These variables constitute the basis for generating weather forecasts in the proposed system, allowing for modeling short-term local climate evolution within the context of the digital twin.

D. *Model Training and Testing*

The weather forecast model was generated following the steps illustrated in Figure 2.

First, the dataset is loaded into a DataFrame, where the temperature, relative humidity, and atmospheric pressure variables are converted to numeric format. The date and time column is transformed into datetime objects and sorted in natural chronological sequence.

Next, a data cleaning process is carried out. Invalid samples are eliminated, including those generated during testing (e.g., records with a frequency of one second) and those with temporal discontinuities greater than one hour.

Once the dataset is cleaned, an exploratory analysis of the variables of interest is performed. To do this, histograms are used to visualize the distribution of the variables extracted from the database.

Subsequently, the input structure for the model is built based on a sliding data window. The input to the forecasting models was defined using a sliding-window approach over the time series. Each input sequence consisted of a window of 24 consecutive hourly samples (corresponding to a 24-hour period), with a stride of 3 hours between windows. This resulted in partially overlapping input sequences, capturing both short-term dynamics and daily patterns. The sampling frequency within each window was fixed at one hour, and the target variable was defined as the meteorological condition in the subsequent 24 hours. Thus, each input had a shape of (24, number of features), where the features corresponded to air temperature, relative humidity, and atmospheric pressure. With this structure, the input tensor is generated, on which statistical analysis is performed to obtain:

- The tensor dimension.
- The total number of elements.
- The number and percentage of null values.
- The percentage of valid data.
- The ranges of values before normalization.
- The identification of potential outliers.

Before training the models, the variables are normalized, and the correlation matrix between them is analyzed to identify relevant relationships between the different meteorological variables.

Different machine learning and deep learning algorithms are trained and comparatively evaluated in the final stage. In the machine learning field, the following models were considered:

- Random Forest
- Gradient Boosting
- Ridge Regression
- Lasso Regression
- K-Nearest Neighbors

The primary evaluation metric employed in this study is the Coefficient of Determination (R^2), which provides a measure of how well the predicted values approximate the actual data. In addition to R^2, the Mean Squared Error (MSE) and Mean Absolute Error (MAE) are also analyzed to gain deeper insights into the model's predictive performance. These complementary metrics allow us to assess aspects such as robustness, generalization capability, and potential overfitting. By considering multiple evaluation criteria, we ensure a more comprehensive understanding of the model's behavior across different scenarios.

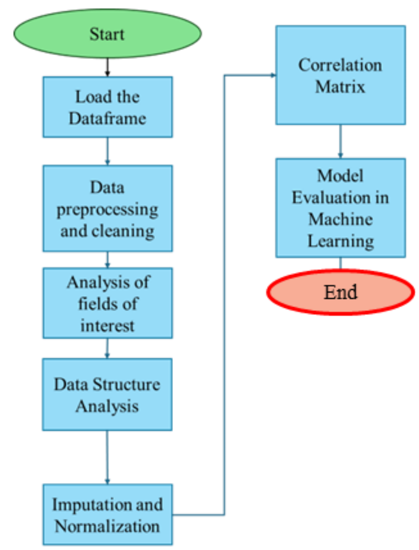

Figure 2. Flowchart of the artificial intelligence model selection, training, and testing process.

E. *Computational Tools and Libraries*

The data processing, analysis, and modeling pipeline was implemented in Python 3.10. For data handling and preprocessing, the libraries Pandas and NumPy were employed to manage time series, perform data cleaning, and generate sliding windows for model input. Exploratory data analysis and visualization were conducted using Matplotlib and Seaborn, which allowed the inspection of statistical distributions, correlations, and temporal trends.

Traditional machine learning models, including Random Forest, Gradient Boosting, Ridge, Lasso, and K-Nearest Neighbors, were implemented with Scikit-learn, which was also used to compute the performance metrics (R^2, MSE, and MAE).

IV. Results

This section will detail the results obtained from the training and testing of the different models proposed for the meteorological dataset.

A. *Histogram of dataset variables*

Figure 3 presents the frequency distributions of the temporal and meteorological variables in the dataset. The temporal variables (minutes, day, and month) display distinct sampling patterns, with the distribution of minutes being nearly uniform across the recorded range, the distribution of days showing alternating frequencies, and the distribution of months indicating data collection concentrated in two main periods.

Regarding the meteorological variables, the air temperature histogram shows a slightly left-skewed distribution, with most observations ranging between 27 °C and 30 °C, and fewer occurrences at extreme values. Relative humidity exhibits a bimodal distribution, with peaks near 70–75 % and around 100 %, indicating frequent saturation events. Atmospheric pressure values are clustered mainly between 1012 hPa and 1016 hPa, following a near-normal distribution with moderate variability.

These histograms provide insight into the statistical characteristics of the dataset, highlighting the predominant conditions recorded by the meteorological station and potential patterns relevant for model training.

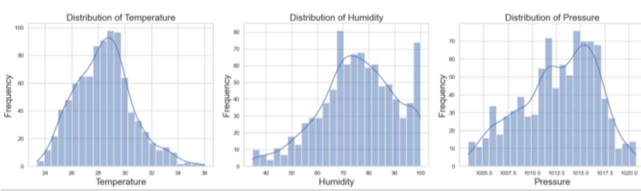

Figure 3. Histrograms of the weather and time variables from the dataset.

B. *Correlation Matrix*

Figure 4 illustrates the relationships between the temporal and meteorological variables in the dataset. When considering the absolute values of the correlation coefficients, the highest associations are observed between relative humidity and both the day and month variables (|r| = 0.60 and 0.51, respectively) negative in sign and between atmospheric pressure and the same variables (|r| = 0.53 and 0.51, respectively) positive in sign. Air temperature also shows a relatively strong correlation (|r| = 0.63) with the minute of the day, indicating a marked diurnal pattern. These high-magnitude correlations, regardless of their direction, suggest that temporal factors exert a significant influence on the measured meteorological variables, and these relationships can be exploited to improve the performance of the forecasting models.

These results indicate the presence of significant relationships among certain variables, which can be exploited to enhance the training process of the artificial intelligence model for local weather forecasting.

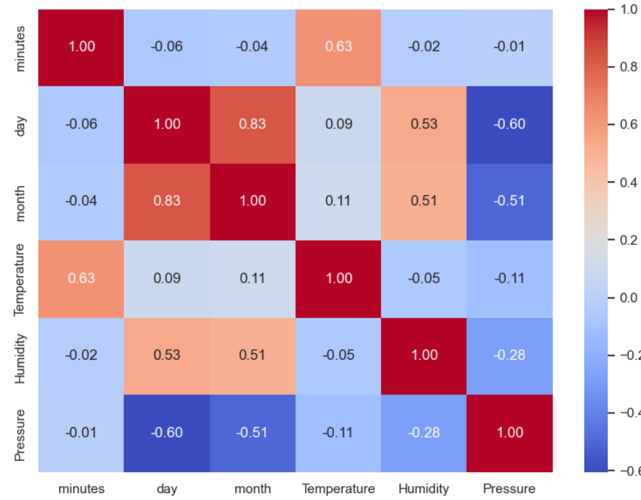

Figure 4. Correlation matrix of temporal and meteorological variables.

C. *Machine learning analysis*

Figure 5 compares the results of the different models for temperature prediction. Gradient Boosting achieved the best overall performance among the classical machine learning methods, with the lowest MAE (0.37) and MSE (0.30), and the highest R^2 (0.656). K-Nearest Neighbors also provided competitive results ($R^2 = 0.640$), followed by Lasso and Ridge regressions with similar performance levels ($R^2 \approx 0.60$). Random Forest yielded the lowest R^2 (0.590) and slightly higher error values.

However, Figure 6 reveals that, despite the relatively high R^2 value, the Gradient Boosting model exhibits substantial dispersion in the test set predictions, not only at extreme temperatures but also in central value ranges. This discrepancy between training and testing performance suggests overfitting, indicating that the model may be capturing noise or dataset-specific patterns rather than generalizable relationships. This limitation will later be addressed in the deep learning experiments, where regularization techniques are introduced to reduce overfitting and improve generalization.

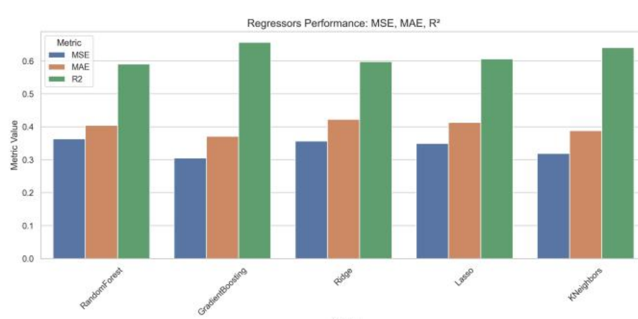

Figure 5. Performance comparison of classical machine learning models for air temperature prediction, evaluated using MSE, MAE, R^2.

 ISBN: 978-1-68558-289-0

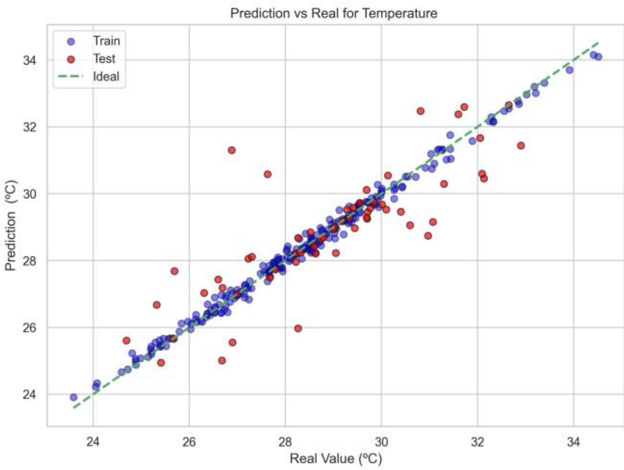

Figure 6. Predicted versus actual air temperature values for the Gradient Boosting model, including both training and testing datasets, compared to the ideal prediction line.

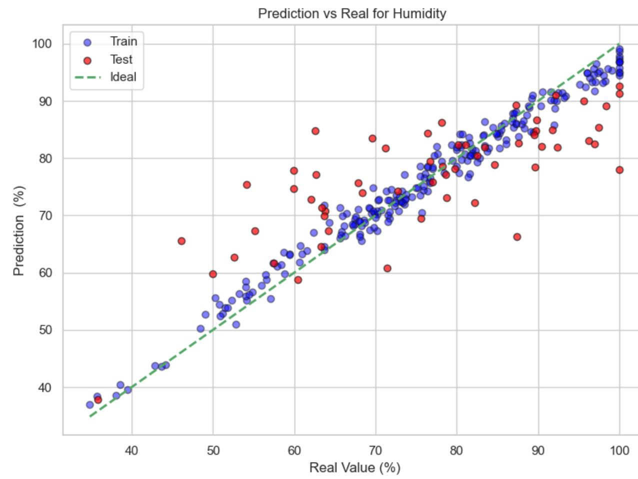

Figure 7. Predicted versus actual relative humidity values for the Gradient Boosting model, including both training and testing datasets, compared to the ideal prediction line.

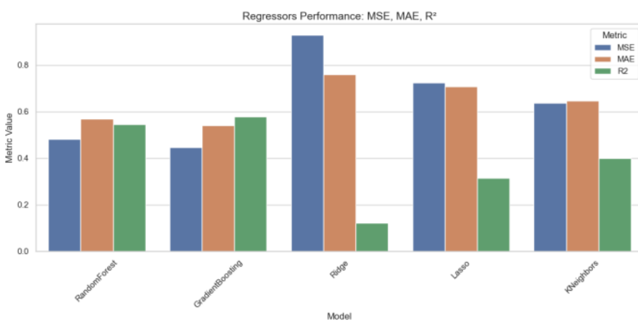

Figure 8. Performance comparison of classical machine learning models for relative humidity prediction, evaluated using MSE, MAE, R^2.

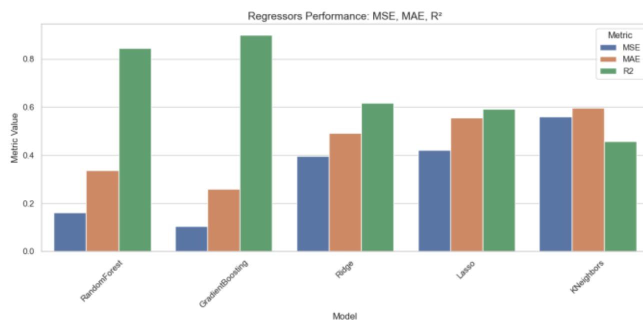

Figure 9. Performance comparison of classical machine learning models for atmospheric pressure prediction, evaluated using MSE, MAE, R^2.

Figure 7 presents the performance comparison for relative humidity prediction. As in the case of temperature, Gradient Boosting outperformed the other models, achieving the highest R^2 (0.578) and the lowest error metrics (MAE = 0.54, MSE = 0.45). Random Forest showed slightly lower performance (R^2 = 0.546), while Ridge Regression performed poorly (R^2 = 0.123), indicating difficulty in modeling the underlying relationships. Lasso Regression and K-Nearest Neighbors produced intermediate results.

Figure 8 depicts the predicted versus actual humidity values for the Gradient Boosting model, showing a strong alignment with the ideal prediction line for most observations, though with higher dispersion in the test set, particularly at mid-range humidity levels.

Figure 9 compares the performance of the evaluated machine learning models for atmospheric pressure prediction. Gradient Boosting achieved the highest coefficient of determination (R^2 = 0.899) and the lowest error metrics (MAE = 0.26, MSE = 0.10), demonstrating its superior accuracy and generalization capacity. Random Forest also performed well, with an R^2 of 0.845 and slightly higher errors (MAE = 0.34, MSE = 0.16). In contrast, Ridge Regression and Lasso Regression yielded moderate results (R^2 = 0.618 and 0.593, respectively), while K-Nearest Neighbors showed the weakest performance (R^2 = 0.457).

Figure 10 illustrates the predicted versus actual atmospheric pressure values using the Gradient Boosting model. The predictions closely follow the ideal line for both training and testing datasets, with minimal dispersion, particularly in the test set. These results confirm that Gradient Boosting provides the most reliable predictions for atmospheric pressure in the given dataset, outperforming all other tested models.

V. Conclusions And Future Work

In this paper, we proposed a system successfully integrates artificial intelligence into a digital twin for local weather forecasting in agricultural environments. The results demonstrate that Gradient Boosting offers the most accurate predictions among the classical machine learning models for all three meteorological variables, with particularly strong performance in atmospheric pressure forecasting.

These outcomes validate the feasibility of combining IoT-based sensing infrastructure with advanced predictive models to enhance decision-making in precision agriculture. By providing localized and timely forecasts, the system can support improved resource allocation, crop management, and environmental sustainability.

 ISBN: 978-1-68558-289-0

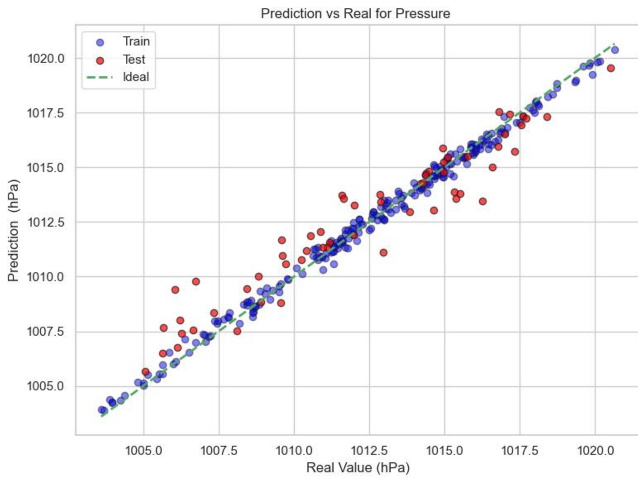

Figure 10. Predicted versus actual atmospheric pressure values for the Gradient Boosting model, including both training and testing datasets, compared to the ideal prediction line.

Future work will focus on expanding the dataset to cover multiple seasons and diverse climatic conditions, integrating additional variables such as wind speed, solar radiation, and soil moisture. Model optimization will include hyperparameter tuning, advanced regularization techniques, and the exploration of hybrid architectures that combine statistical and neural approaches. Furthermore, deploying the forecasting system in real-time operational scenarios and integrating it with automated control mechanisms in the digital twin will be key steps towards its practical adoption in smart farming applications.

ACKNOWLEDGMENT

This work is funded by the Ministry of Science, Innovation and Universities through the State Research Agency (AEI) with the projects PID2020-114467RR-C33/AEI/10.13039/501100011033, TED2021-131040B-C31.

REFERENCES

[1] L. Wang, "Digital twins in agriculture: a review of recent progress and open issues," *Electronics*, vol. 13, no. 11, p. 2209, 2024.

[2] J. Lloret, M. Garcia, D. Bri, and J. R. Diaz, "A cluster-based architecture to structure the topology of parallel wireless sensor networks," *Sensors*, vol. 9, no. 12, pp. 10513–10544, 2009.

[3] A. Hussein, A. Challoob, and M. Yousif, "Optimizing network topologies for enhanced communication and scalability in distributed networks," *J. Sci. Res. Teach.*, vol. 3, no. 11, 2024.

[4] W. Purcell, T. Neubauer, and K. Mallinger, "Digital twins in agriculture: challenges and opportunities for environmental sustainability," *Curr. Opin. Environ. Sustain.*, vol. 61, p. 101252, 2023.

[5] P. Angin, M. H. Anisi, F. Göksel, C. Gürsoy, and A. Büyükgülcü, "Agrilora: a digital twin framework for smart agriculture," *J. Wirel. Mob. Netw. Ubiquitous Comput. Dependable Appl.*, vol. 11, no. 4, pp. 77–96, 2020.

[6] N. Peladarinos, D. Piromalis, V. Cheimaras, E. Tserepas, R. A. Munteanu, and P. Papageorgas, "Enhancing smart agriculture by implementing digital twins: A comprehensive review," *Sensors*, vol. 23, no. 16, p. 7128, 2023.

[7] H. Zhang, Y. Liu, C. Zhang, and N. Li, "Machine learning methods for weather forecasting: A survey," *Atmosphere*, vol. 16, no. 1, p. 82, 2025.

[8] R. Ahsen et al., "Harnessing digital twins for sustainable agricultural water management: A systematic review," *Appl. Sci.*, vol. 15, no. 8, p. 4228, 2025.

[9] M. Awais, X. Wang, S. Hussain, F. Aziz, and M. Q. Mahmood, "Advancing precision agriculture through digital twins and smart farming technologies: A review," *AgriEngineering*, vol. 7, no. 5, p. 137, 2025.

[10] A. Khanna and S. Kaur, "Evolution of Internet of Things (IoT) and its significant impact in the field of precision agriculture," *Comput. Electron. Agric.*, vol. 157, pp. 218–231, 2019.

[11] N. Ariesen-Verschuur, C. Verdouw, and B. Tekinerdogan, "Digital twins in greenhouse horticulture: A review," *Comput. Electron. Agric.*, vol. 199, p. 107183, 2022.

[12] J. A. López, A. J. Garcia-Sanchez, F. Soto, A. Iborra, F. Garcia-Sanchez, and J. Garcia-Haro, "Design and validation of a wireless sensor network architecture for precision horticulture applications," *Precis. Agric.*, vol. 12, no. 2, pp. 280–295, 2011.

[13] D. Bri, M. Garcia, J. Lloret, and P. Dini, "Real deployments of wireless sensor networks," in *Proc. 2009 3rd Int. Conf. Sensor Technol. Appl. (Sensorcomm 2009)*, Athens, Greece, Jun. 2009, pp. 415–423.

[14] M. M. Sakha, F. Daiber, C. Tieben, and M. Enders, "Virtual breeding nursery: Towards a VR digital twin for plant breeding," in *Proc. 2024 ACM Symp. Spatial User Interact.*, 2024, pp. 1–2.

[15] O. Spyrou, M. Ariza-Sentís, and S. Vélez, "Enhancing education in agriculture via XR-based digital twins: A novel approach for the next generation," *Appl. Syst. Innov.*, vol. 8, no. 2, p. 38, 2025.

[16] M. Awais, X. Wang, S. Hussain, F. Aziz, and M. Q. Mahmood, "Advancing precision agriculture through digital twins and smart farming technologies: A review," *AgriEngineering*, vol. 7, no. 5, p. 137, 2025.

[17] M. Garcia, S. Sendra, J. Lloret, and A. Canovas, "Saving energy and improving communications using cooperative group-based wireless sensor networks," *Telecommun. Syst.*, vol. 52, no. 4, pp. 2489–2502, 2013.

[18] Y. Zhang, J. Li, X. Wang, and P. Zhao, "Deep learning-based environmental monitoring: A survey," *Environ. Model. Softw.*, vol. 156, p. 105495, 2022.

[19] S. Cesco, P. Sambo, M. Borin, B. Basso, G. Orzes, and F. Mazzetto, "Smart agriculture and digital twins: Applications and challenges in a vision of sustainability," *Eur. J. Agron.*, vol. 146, p. 126809, 2023.

[20] A. A. Hapsari, D. J. Vresdian, M. Aldiansyah, B. W. Dionova, and A. C. Windari, "Indoor air quality monitoring system with Node.js and MQTT application," in *Proc. 2020 1st Int. Conf. Inf. Technol., Adv. Mech. Electr. Eng. (ICITAMEE)*, Yogyakarta, Indonesia, 2020, pp. 144–149.

 ISBN: 978-1-68558-289-0