European Journal of Sustainable Development (2026), **15**, 4, 655-680 ISSN: 2239-5938
Doi: 10.14207/ejsd.2026.v15n4p655

An Intelligent Fog–Cloud Smart Irrigation System with Hybrid Machine Learning and Evapotranspiration-Based Control

By Mohammed Balfaqih[1], Atheer Nagro[1], Abdulaziz Al-Tuwairqi[1]

ABSTRACT:
Sustainable agricultural water management is increasingly important in arid and semi-arid regions, where limited freshwater resources, climatic variability, and inefficient irrigation practices can constrain long-term food production. This study proposes an intelligent fog–cloud smart irrigation framework that integrates Internet of Things sensing, LoRa/LoRaWAN communication, evapotranspiration-based irrigation calculation, machine-learning prediction, and cloud-assisted historical search within a hierarchical decision architecture. The system prioritizes local fog-layer processing for routine irrigation decisions, while cloud resources are reserved for long-term storage and fallback support when sensor information is incomplete or local predictions are insufficiently reliable. A K-nearest-neighbor classifier was evaluated using 104 site-day observations with stratified five-fold cross-validation and achieved 71.2% accuracy, outperforming the evaluated Random Forest, Logistic Regression, Decision Tree, and SVM-RBF models in several performance measures. Prototype testing verified the end-to-end operation of sensing, irrigation actuation, water-flow monitoring, and remote visualization. Using the recorded 52-day sensing dataset, the adaptive irrigation strategy reduced combined water use by 58.7% relative to a simulated daily fixed-schedule baseline and by 17.3% relative to an every-two-day schedule. Simulation-based system evaluation further indicated a 95.2% reduction in mean decision latency for fog processing compared with cloud-centered processing, while modeled communication overhead decreased by up to 87.5% in message count and 75.0% in transmitted data volume under complete sensor availability. Missing-sensor experiments demonstrated that the proposed fallback mechanism maintains decision continuity by progressively shifting uncertain cases from evapotranspiration calculation to K-NN prediction and cloud-assisted retrieval. The proposed framework demonstrates the potential of combining adaptive irrigation, distributed intelligence, and reliability-aware decision making to support more resource-efficient and sustainable agricultural water management.

Keywords: sustainable agriculture, Smart irrigation, Water conservation, Fog computing, Internet of Things, Machine learning, Evapotranspiration, LoRaWAN

1. Introduction

Agriculture is a major consumer of freshwater resources, making irrigation efficiency particularly important in arid and semi-arid regions where water scarcity constrains agricultural sustainability. In Saudi Arabia, climatic and geographical conditions intensify pressure on available water resources, while alternatives such as reclaimed water and desalinated seawater involve concerns related to safety, energy requirements, and cost (Gonzales-Gustavson et al., 2019; Martínez-Alvarez et al., 2016). Improving irrigation

[1]Department of Computer and Network Engineering, University of Jeddah, Jeddah 23890, Saudi Arabia.

efficiency is therefore important not only for freshwater conservation but also for sustaining agricultural productivity and food security.

Conventional irrigation practices often rely on fixed schedules, uniform water delivery, or farmer experience without continuously accounting for actual soil and environmental conditions. Such practices can result in over-irrigation or poorly timed irrigation, increasing water consumption and adversely affecting soil and root conditions (Chen et al., 2011). Excessive irrigation may also contribute to salinity accumulation and degradation of soil fertility (Shrivastava & Kumar, 2015). Precision agriculture provides a framework for improving resource-use efficiency through sensing, data analysis, and automated control (Zhang et al., 2002). Smart irrigation extends this approach by adapting water delivery to crop and environmental requirements rather than relying exclusively on predetermined schedules (Brouwer & Heibloem, 1986; Calera et al., 2017; Kashyap et al., 2021; Mohammed et al., 2020).

The Internet of Things (IoT) has become an important enabling technology for smart irrigation because distributed sensors can continuously acquire soil, climatic, and operational information. Previous systems have incorporated soil-moisture, temperature, humidity, weather, and other environmental measurements to support irrigation monitoring and control (Goap et al., 2018; Gutiérrez et al., 2014; Salvi et al., 2017; Saqib et al., 2020; Tzounis et al., 2017). However, many approaches remain dependent on fixed thresholds, a limited set of environmental parameters, or centralized cloud processing. These characteristics may limit adaptation to interactions among crop type, soil condition, atmospheric demand, and irrigation-system characteristics. Continuous transmission of field measurements to remote cloud resources may also increase communication overhead and dependence on network availability, particularly in large-scale or remote agricultural environments.

Machine learning (ML) has increasingly been investigated to capture nonlinear relationships among environmental variables and irrigation requirements (Goap et al., 2018; Kashyap et al., 2021; Torres-Sánchez et al., 2020). Nevertheless, the effectiveness of ML-based irrigation depends on the representativeness and completeness of the available data, while sensor failures or missing measurements can reduce decision reliability. Recent research has consequently emphasized hybrid decision mechanisms and distributed edge/fog processing to improve responsiveness and reduce dependence on centralized cloud resources (Bautista et al., 2026; Liu et al., 2026).

A preliminary version of this work presented an IoT-enabled irrigation prototype integrating environmental sensing, ML-based irrigation decisions, automatic pump control, and remote monitoring (Balfaqih, 2023). The preliminary study established the feasibility of the basic sensing-to-actuation workflow but did not incorporate distributed fog–cloud processing, evapotranspiration-based irrigation calculation, long-range LoRa/LoRaWAN communication, or a reliability-aware fallback mechanism. The present study substantially extends that system through a hierarchical fog–cloud architecture that integrates crop-specific soil-moisture boundaries, FAO Penman–Monteith evapotranspiration estimation, K-nearest-neighbor (K-NN) prediction, and cloud-assisted historical retrieval. The main contributions of this work are summarized as follows:

- A hierarchical fog–cloud smart irrigation architecture is developed by integrating LoRa/LoRaWAN field communication, distributed fog-based processing, and cloud-assisted historical analysis.
- A hybrid irrigation decision mechanism is proposed by combining crop-specific soil-moisture boundaries, FAO Penman–Monteith evapotranspiration calculation, K-NN-based irrigation prediction, and k-d-tree-based historical search.
- A reliability-aware irrigation workflow is introduced in which missing or incomplete sensor information can trigger alternative decision routes, reducing dependence on a single data source or prediction method.
- The system dynamically estimates irrigation volume and duration using crop, environmental, and flow-rate information, while supporting remote monitoring and user interaction through the application layer.

The remainder of the paper is organized as follows. Section 2 reviews related work on IoT-enabled irrigation, machine-learning-based decision making, and fog–cloud agricultural systems. Section 3 presents the proposed irrigation architecture and its decision mechanisms. Section 4 describes the implementation and evaluation methodology. Section 5 presents and discusses the machine-learning, prototype, water-consumption, latency, reliability, and communication-overhead results. Section 6 concludes the study and outlines future research directions.

2. Related work

Smart irrigation systems have evolved from basic threshold-based controllers toward IoT-enabled platforms that integrate environmental sensing, automated irrigation, remote monitoring, and intelligent decision support. Previous studies have combined soil-moisture, temperature, humidity, weather, and other field measurements with local or cloud-based monitoring to support irrigation scheduling and control (Goap et al., 2018; Gutiérrez et al., 2014; Salvi et al., 2017; Tzounis et al., 2017). These approaches demonstrate the practical value of continuous field observation; however, many remain dependent on fixed thresholds, centralized cloud processing, or limited decision logic, which can constrain adaptability, communication efficiency, and deployment flexibility.

Recent studies have focused on improving water-use efficiency through low-cost IoT platforms and more advanced irrigation indicators. Boukri et al. (2026) developed an ESP32-based irrigation system using soil-moisture and environmental sensing and reported water savings of approximately 25–35% compared with fixed-schedule irrigation. Mohamed et al. (2026) reported a 47% reduction in water consumption using an IoT-based irrigation framework integrated with fertilization and biopesticide management, although the specific contribution of the irrigation strategy cannot be fully separated from the additional agronomic interventions. Morchid et al. (2026) combined ESP32-based sensing with the ThingsBoard cloud platform for automated irrigation and remote monitoring, while Ayat et al. (2026) incorporated LoRa communication, FAO-56 reference evapotranspiration, and the Crop Water Stress Index within a low-power precision-irrigation system. Taken together, these studies show that combining field sensing with low-power wide-area connectivity and crop-water-demand indicators can improve

© 2026 The Authors. Journal Compilation © 2026 European Center of Sustainable Development.

automated irrigation management. Their control strategies, however, generally rely on deterministic rules, cloud-centered operation, or one principal decision mechanism.

Machine learning has increasingly been employed to capture nonlinear relationships among soil, climatic, and irrigation variables. Goap et al. (2018) applied ML techniques to irrigation prediction using soil and environmental measurements, while Dutta et al. (2014) developed an intelligent environmental knowledge system for agricultural decision support. Torres-Sánchez et al. (2020) compared several learning techniques for irrigation management and reported prediction errors below 10% relative to agronomist recommendations. More recent studies have combined ML with IoT-enabled irrigation platforms. Rahul et al. (2026) evaluated multiple classifiers within an IoT-based irrigation system and reported a 35.1% reduction in water consumption, whereas Rani and Banu (2026) reported high classification performance using a feed-forward learning model. Despite these advances, ML-based irrigation remains sensitive to limited datasets, sensor failures, missing observations, and dependence on continuous communication with cloud resources (Bautista et al., 2026; Liu et al., 2026).

Fog and edge computing have therefore received increasing attention as mechanisms for reducing latency, communication overhead, and dependence on centralized cloud processing. By moving part of the decision process closer to the field, routine irrigation operations can be performed locally while cloud resources are reserved for long-term storage, historical analysis, and additional computation (Bautista et al., 2026; Liu et al., 2026). However, relatively few smart irrigation systems combine physically based irrigation estimation, local ML prediction, and cloud-assisted historical retrieval as complementary decision mechanisms. The preliminary system presented by Balfaqih (2023), for example, demonstrated an IoT-enabled irrigation prototype with ML-based classification, automatic pump control, and remote monitoring, but it did not incorporate fog–cloud processing, evapotranspiration-based calculation, or a reliability-aware fallback mechanism.

The present work addresses this gap through a hierarchical architecture that combines LoRa/LoRaWAN communication, fog-based processing, crop-specific soil-moisture limits, FAO Penman–Monteith evapotranspiration calculation, K-NN-based irrigation prediction, and cloud-assisted historical search. Rather than depending exclusively on fixed thresholds, a single ML model, or continuous cloud processing, the proposed framework selects the appropriate decision path according to sensor-data availability and prediction confidence. Table 1 summarizes representative smart irrigation systems and compares them in terms of processing architecture, irrigation decision strategy, long-range communication, and missing-data or fallback handling.

Table 1: Architectural and decision-mechanism characteristics of selected smart irrigation approaches.

Study	Processing architecture	Decision approach	Long-range communication	Missing-data/fallback handling	Main limitation
Boukri et al. (2026)	ESP32 + cloud/app	Fixed moisture thresholds	No	No	Limited adaptation to crop/environment variability

Rahul et al. (2026)	IoT + cloud	Multiple ML classifiers	No	Anomaly filtering only	Cloud remains part of end-to-end pipeline
Ayat et al. (2026)	IoT + LoRa	ET0 + CWSI + soil indicators	Yes	No	No ML or historical fallback
Morchid et al. (2026)	ESP32 + ThingsBoard cloud	Automated rule-based control	No	No	Connectivity and calibration dependence
Balfaqih (2023)	IoT + local microcontroller	ML classification	No	No	No hierarchical fallback or fog–cloud integration
Proposed system	LoRa/LoRaWAN + fog + cloud	ET calculation + K-NN prediction + k-d tree search	Yes	Yes	Requires broader field-scale validation

3. Proposed fog–cloud intelligent smart irrigation system

The proposed smart irrigation framework extends the preliminary system presented by Balfaqih (2023) through a hierarchical fog–cloud architecture that integrates long-range field communication, local decision processing, and cloud-assisted fallback. The following subsections describe the architecture, sensing requirements, decision workflow, and irrigation-control mechanisms.

3.1 System architecture

The overall architecture of the proposed irrigation system consists of four main layers: acquisition, communication, processing, and application. Figure 1 illustrates the four-layer architecture of the proposed system. The acquisition layer collects soil moisture, soil temperature, air temperature, relative humidity, solar radiation, wind speed, and water-flow measurements. These data are transmitted through LoRa/LoRaWAN to the communication gateway and forwarded to nearby fog-processing resources. Routine irrigation decisions are performed locally at the fog layer, while the cloud supports long-term storage, historical analysis, and fallback retrieval when additional decision support is required. The application layer provides web- and mobile-based access to sensor measurements, irrigation status, and control information. The architecture also supports zone-based farm management, allowing areas with different crop and soil conditions to be monitored and controlled independently within the same agricultural field.

© 2026 The Authors. Journal Compilation © 2026 European Center of Sustainable Development.

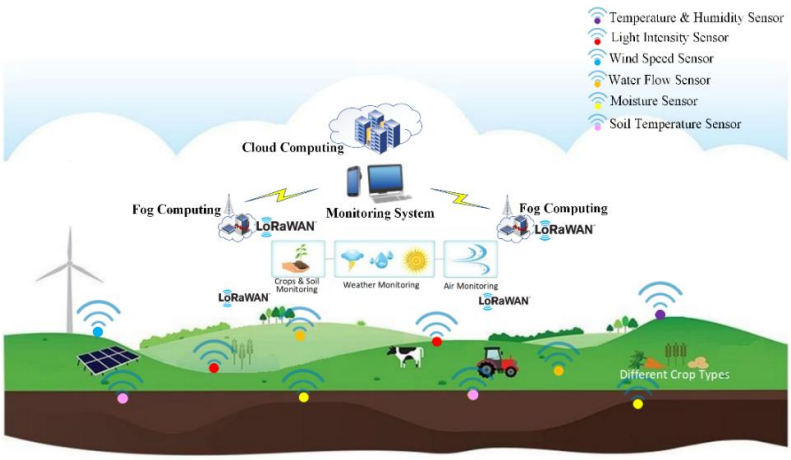

Figure 1: Architecture of the Proposed Smart Irrigation System (adapted from Balfaqih, 2023).

3.2 Sensing and data acquisition

The irrigation decision combines real-time sensor measurements with predefined farm information. Field conditions are characterized through measurements of soil-water status, soil and ambient temperature, atmospheric humidity, incident solar radiation, wind velocity, and irrigation flow. Crop type, soil characteristics, farm location, zone configuration, and irrigation-rate information are supplied as predefined management parameters.

Crop-specific soil-moisture limits are used to determine the initial irrigation requirement. Rather than applying a single universal threshold, the proposed system defines a minimum acceptable soil-moisture level, $SM^{M}_{\min}$, and a maximum level, $SM^{M}_{\max}$, according to the crop type. Irrigation is initiated when the measured soil moisture falls below the minimum boundary and continues until the maximum boundary is reached. For the i-th day, irrigation is initiated when

$$S_i^M \leq SM^M_{\min} \tag{1}$$

and continues until

$$S_i^M \geq SM^M_{\max}. \tag{2}$$

Using a lower and upper boundary rather than a single threshold reduces repeated switching of the relay and associated actuator components when the measured soil moisture fluctuates around one decision point.

The fog node also checks whether the expected sensor readings have been received. If one or more readings are missing, the user is notified so that the corresponding sensor can be inspected or replaced. Rather than terminating the irrigation decision process, the system can then move to an alternative prediction or cloud-assisted decision route, as described in the following subsections.

3.3 Fog-based irrigation decision and control

The detailed operational workflow of the proposed system is presented in Figure 2. The sequence diagram shows the interaction among the field sensors, fog node, cloud server, user platform, and irrigation actuator.

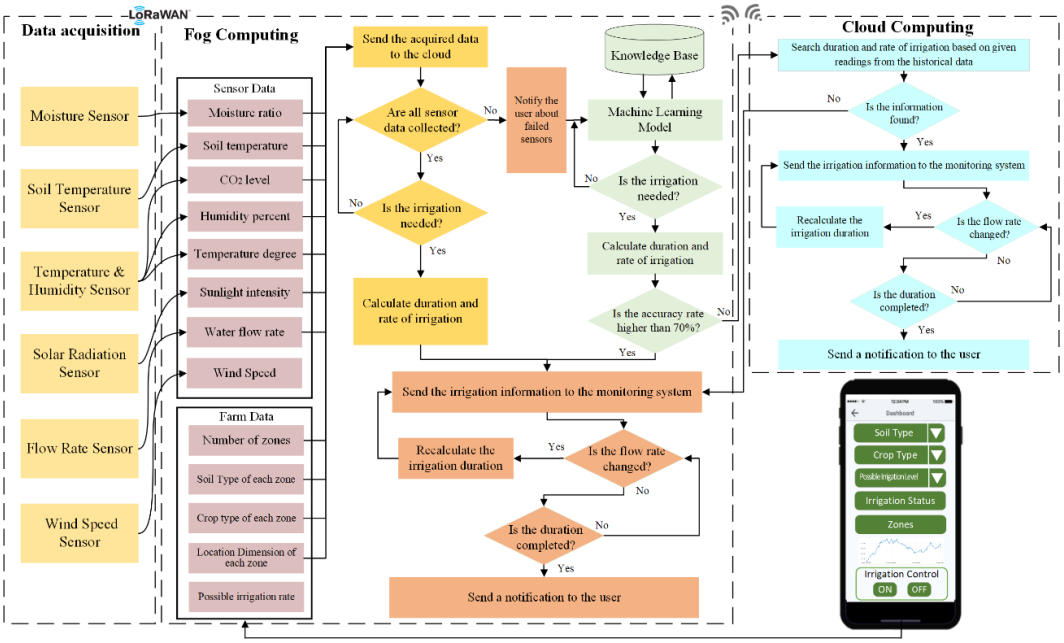

Figure 2: Detailed sequence diagram of the proposed irrigation system.

At each decision cycle, the fog node receives the current sensor measurements together with the relevant crop and farm parameters. Irrigation need is first determined using the crop-specific soil-moisture boundaries defined in Section 3.2. When irrigation is required, the system selects one of three decision routes according to data availability and prediction reliability. Complete measurements are processed using the evapotranspiration-based Irrigation Calculation Method; incomplete measurements trigger the K-NN-based Irrigation Prediction Method; and predictions below the predefined confidence threshold are escalated to the cloud-based Irrigation Search Method.

Once the required irrigation volume and duration are determined, the resulting command is applied to the irrigation actuator and reported through the user platform. The water-flow sensor monitors the delivered flow during irrigation, allowing the irrigation duration to be adjusted when the measured flow rate changes

3.4 Irrigation calculation method

When the environmental measurements required for physically based irrigation estimation are available, the fog node determines the crop water requirement using evapotranspiration. Crop evapotranspiration, ET_c, is calculated using the crop-coefficient approach:

$$ET_c = K_c ET_0 \tag{3}$$

© 2026 The Authors. Journal Compilation © 2026 European Center of Sustainable Development.

where K_c is the crop coefficient, which varies according to crop type and growth stage, and ET_0 is the reference evapotranspiration in mm/day. Reference evapotranspiration is calculated using the FAO Penman–Monteith formulation (Allen et al., 1998):

$$ET_0 = \frac{0.408\Delta(R_n - G) + \gamma\left(\frac{900}{T+273}\right)u_2(e_s - e_a)}{\Delta + \gamma(1 + 0.34u_2)} \quad (4)$$

where R_n is the net radiation at the crop surface, G is the soil heat flux, T is the mean air temperature, $u_2$is the wind speed at 2 m height, e_s and e_a are the saturation and actual vapor pressures, respectively, Δ is the slope of the saturation vapor-pressure curve, and γ is the psychrometric constant. The supporting meteorological parameters, including vapor-pressure terms, wind-speed adjustment, atmospheric pressure, psychrometric constant, and vapor-pressure slope, are calculated according to the standard FAO-56 procedure. For daily calculations, soil heat flux is assumed to be negligible $(G \approx 0)$.

After ET_c is determined, the required irrigation volume is calculated as

$$IV = \frac{0.5\, ET_c A_s}{1000} \quad (5)$$

where A_s is the total irrigated area and the factor 0.5 represents the assumed wetted-area fraction, corresponding to 50% of the total surface area in the present implementation. The required irrigation duration is subsequently obtained from the measured water-flow rate:

$$t_{\text{irr}} = \frac{IV}{Q} \quad (6)$$

where Q denotes the measured water-flow rate. Because the flow rate is monitored during irrigation, the irrigation duration can be recalculated when Q changes, allowing the system to maintain the intended delivered water volume.

3.5 Machine learning prediction method

When the complete set of measurements required for the evapotranspiration-based calculation is unavailable, the fog node switches to the Irrigation Prediction Method. This method employs the K-nearest-neighbor (K-NN) algorithm using the dataset described in Section 4.

The K-NN model is used to determine whether irrigation is required when the complete set of measurements needed for the evapotranspiration-based calculation is unavailable. For a current feature vector x, the model identifies the k most similar observations in the training dataset according to their standardized feature-space distance. The irrigation decision is then determined from the class labels of the neighboring observations. The neighborhood size $k$is selected empirically using cross-validation, as described in Section 4.3.

The proposed framework applies a prediction-confidence threshold of 70% to the K-NN output. When the confidence associated with the predicted irrigation class is at least 70%, the local prediction is accepted by the fog node. Predictions below this threshold are treated as insufficiently reliable and are therefore escalated to the cloud-based Irrigation Search Method. The 70% value is used in the present study as a predefined operating

threshold and its influence on the reliability–communication trade-off is examined in Section 5.3.3. The K-NN prediction stage therefore serves as an intermediate decision mechanism between direct physical calculation and historical cloud retrieval. Its role is particularly important when sensor failures or missing measurements prevent the complete evapotranspiration calculation.

3.5 Cloud-based historical search and reliability mechanism

The cloud server maintains historical records describing plant characteristics, environmental conditions, and irrigation operations collected from connected farms. When additional decision support is required, the cloud first searches records from the same farm and can subsequently extend the search to farms in the same city or in locations with similar climatic and crop conditions. A k-d tree is used to organize the multidimensional historical dataset and efficiently retrieve records with similar soil, crop, environmental, and irrigation characteristics (Mehlhorn, 2013). When a sufficiently similar historical case is identified, its irrigation volume and duration are returned to support the current decision. If no suitable match is found, the previously generated ML prediction is retained. The overall decision hierarchy is summarized in Table 2. Complete sensor information is processed using the evapotranspiration-based calculation, incomplete information is handled through K-NN prediction, and predictions that do not satisfy the predefined confidence criterion are escalated to cloud-based historical search.

Table 2: Decision hierarchy of the proposed irrigation system

System condition	Decision method	Processing location
Complete required sensor information is available	Evapotranspiration-based irrigation calculation	Fog node
Required sensor information is incomplete	K-NN-based irrigation prediction	Fog node
ML prediction does not satisfy the acceptance criterion	Historical similarity search using k-d tree	Cloud server
No suitable historical match is identified	Use the available ML prediction	Fog node

4. Implementation and evaluation

This section describes the implementation of the proposed fog–cloud irrigation framework and the methodology used to evaluate its machine-learning and system-level performance. The implementation integrates field sensing, LoRa-based communication, fog-side decision processing, cloud-assisted storage and historical search, irrigation actuation, and remote monitoring. The evaluation considers both the predictive performance of the irrigation decision models and the operational behavior of the proposed architecture under representative irrigation scenarios.

4.1 Prototype and communication implementation

The prototype was designed to realize the main components of the architecture presented in Figure 1, including the sensing, communication, processing, and application layers. The field sensing subsystem acquires the environmental and irrigation variables

© 2026 The Authors. Journal Compilation © 2026 European Center of Sustainable Development.

required by the proposed decision process, including soil moisture, soil temperature, air temperature, relative humidity, solar radiation, wind speed, and water-flow rate. These measurements are collected at the field level and forwarded to the communication subsystem for further processing.

LoRa communication is employed between the sensing nodes and the LoRa gateway to provide long-range data transmission with relatively low communication overhead. The gateway acts as an intermediate communication point between the distributed field nodes and the higher processing layers. Sensor measurements received by the gateway are forwarded to the fog-processing node, while cellular connectivity is used to communicate with the cloud server when remote storage, historical analysis, or cloud-assisted search is required. This communication structure follows the layered architecture illustrated in Figure 1 and supports multi-zone farm management with independent sensing and irrigation control across different field areas.

The fog node performs the time-sensitive irrigation functions described in Section 3. It receives the current sensor readings together with predefined farm parameters, including crop type, soil type, farm location, number of zones, and possible irrigation rate. Based on the availability of the required data, the fog node either performs the evapotranspiration-based irrigation calculation or invokes the K-NN prediction model. When the prediction does not meet the predefined acceptance criterion, the fog node communicates with the cloud server to initiate the historical-search procedure.

It should be noted that the physical prototype primarily validates the sensing, local processing, actuation, water-flow measurement, and remote-monitoring functions inherited from the preliminary implementation. The extended multi-zone LoRa/LoRaWAN fog–cloud architecture represents the journal-level system design, while its latency, missing-sensor behavior, and communication overhead are evaluated through simulation in Section 5.3. Accordingly, the prototype and simulation results are treated as complementary levels of validation rather than as evidence of a complete field-scale LoRaWAN deployment.

The cloud server stores the environmental, crop, and irrigation records generated by the system and supports the k-d-tree-based historical search mechanism. The stored data can also be used for long-term analysis, including examination of seasonal irrigation patterns and irrigation quantities. The application layer provides remote visualization of sensor readings, irrigation status, and recommended irrigation information and allows the user to interact with the irrigation actuator when required.

The water-flow sensor provides feedback during irrigation. Once the required irrigation volume has been determined, the irrigation duration is calculated from the measured flow rate. If the flow rate changes during an irrigation event, the duration is recalculated accordingly to maintain the intended delivered water volume.

4.2 Irrigation dataset and data processing

The machine-learning evaluation was conducted using daily aggregated observations obtained from the SA01 and SAP01 sensing units over 52 consecutive calendar days, from 2 March to 22 April 2021. Each site-day was treated as one observation, resulting in a total of 104 labeled samples. The irrigation-decision label was derived from the recorded daily water-flow volume: observations with a recorded irrigation

volume greater than 0 L were labeled as *irrigation required*, whereas observations with no recorded water flow were labeled as *no irrigation required*. The resulting dataset contained 43 irrigation-required observations (41.3%) and 61 no-irrigation observations (58.7%).

The predictor set consisted of eight environmental and evapotranspiration-related variables available for both sensing sites: air temperature, soil temperature, relative humidity, solar radiation measured as photosynthetically active radiation (PAR), wind speed, FAO Penman–Monteith reference evapotranspiration, Hargreaves–Samani reference evapotranspiration, and Penman–Monteith reference evapotranspiration. Because K-NN relies on distance calculations, feature standardization was applied to prevent variables with larger numerical ranges from disproportionately influencing neighborhood selection.

Given the relatively small dataset, stratified five-fold cross-validation was adopted instead of a single holdout split, with the class distribution preserved across folds. To maintain a leakage-free evaluation protocol, the standardization parameters were estimated independently using only the training data within each fold and were subsequently applied to the corresponding validation data.

4.3 Machine learning evaluation

The machine-learning evaluation assessed whether K-NN can provide an auxiliary irrigation decision when some of the measurements required for the evapotranspiration-based calculation are unavailable. Candidate neighborhood sizes from $K = 1$ to $K = 15$ were evaluated using stratified five-fold cross-validation, and the value providing the highest cross-validated classification accuracy was selected.

To determine whether K-NN provided competitive predictive performance, four additional classifiers were evaluated under the same cross-validation protocol: Decision Tree, Random Forest with 200 trees, Support Vector Machine with a radial basis function kernel, and Logistic Regression. Performance was quantified using Accuracy, Precision, Recall, weighted F1-score, Cohen's Kappa, ROC-AUC, and the confusion matrix. The principal classification metrics are defined as

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \tag{7}$$

$$\text{Precision} = \frac{TP}{TP + FP} \tag{8}$$

$$\text{Recall} = \frac{TP}{TP + FN} \tag{9}$$

and

$$F_1 = 2\frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{10}$$

Cohen's Kappa was additionally used to quantify agreement beyond chance, while ROC-AUC was used to assess class discrimination independently of the selected classification threshold. The confusion matrix was examined to identify class-specific errors, particularly false-negative irrigation decisions.

In addition to the global classification metrics, the operational decision workflow applies a 70% prediction-confidence threshold. K-NN predictions below this confidence

© 2026 The Authors. Journal Compilation © 2026 European Center of Sustainable Development.

level are not accepted locally and are instead escalated to the cloud-assisted historical-search mechanism.

4.4 Experimental test scenario

The physical prototype was evaluated through a ten-day small-scale irrigation experiment designed to verify the complete sensing-to-decision-to-actuation workflow. During operation, the system acquired environmental and soil measurements, processed the irrigation decision locally, controlled the irrigation pump through the relay interface, recorded the delivered water volume using the flow sensor, and transmitted the monitored information to the user interface. The experiment was used primarily to validate functional integration among sensing, processing, actuation, water-flow measurement, and remote monitoring.

In addition to the physical prototype experiment, the recorded 52-day sensing dataset from the SA01 and SAP01 units was used for machine-learning and water-consumption analysis. System-level characteristics that were not directly timestamped during the preliminary deployment, including fog-versus-cloud latency, missing-sensor behavior, and communication overhead, were subsequently evaluated through simulation as described in Section 5.3. The experimental and simulation results are therefore reported separately to distinguish physically observed behavior from modeled system performance.

4.5 System performance metrics

In addition to machine-learning performance, the proposed fog–cloud architecture was evaluated using four system-level criteria: irrigation water consumption, decision latency, communication overhead, and reliability under missing-sensor conditions. Water consumption was quantified using the recorded water-flow information, whereas latency, communication overhead, and missing-sensor behavior were evaluated through simulation because the preliminary prototype did not provide the timestamped communication and processing measurements required for direct system-level measurement. When a conventional or fixed-schedule irrigation baseline is available, the percentage water saving can be calculated as

$$\text{Water Saving}(\%) = \frac{V_{\text{baseline}} - V_{\text{proposed}}}{V_{\text{baseline}}} \times 100 \tag{11}$$

where V_{baseline} is the total irrigation volume used by the baseline method and V_{proposed} is the volume consumed by the proposed system.

The irrigation decision latency can be defined as the elapsed time between acquisition of the sensor information and generation of the corresponding irrigation command:

$$T_{\text{decision}} = t_{\text{command}} - t_{\text{sensing}} \tag{12}$$

To quantify the benefit of fog-side processing, the fog decision latency can be compared with a cloud-processing scenario. The relative latency reduction can then be expressed as

$$\text{Latency Reduction}(\%) = \frac{T_{\text{cloud}} - T_{\text{fog}}}{T_{\text{cloud}}} \times 100 \tag{13}$$

Communication overhead was quantified using the number of transmitted messages and the corresponding transmitted data volume over a predefined observation period. This measure is particularly relevant because the proposed architecture is intended to perform routine irrigation processing locally while using the cloud only when additional analysis or historical retrieval is required. Finally, the reliability of the hierarchical decision mechanism was evaluated by introducing controlled levels of sensor-data missingness and recording the proportion of irrigation requests resolved through evapotranspiration calculation, accepted K-NN prediction, or cloud-based historical search.

5. Results and discussion

The evaluation considers three complementary aspects: machine-learning performance, prototype functionality, and system-level performance. The ML component is assessed using cross-validated classification metrics, while the prototype is evaluated for end-to-end sensing, processing, actuation, and monitoring. System-level performance is examined in terms of water consumption, decision latency, resilience to missing sensor data, and communication overhead. Water-consumption results are based on recorded data, whereas latency, missing-sensor behavior, and communication overhead are evaluated through simulation.

5.1 Machine learning performance

The K-NN irrigation-decision classifier described in Section 3.5 was evaluated using the 104 site-day observations and the stratified five-fold cross-validation protocol described in Sections 4.2 and 4.3. The model was assessed as an auxiliary decision mechanism for situations in which the complete measurements required for evapotranspiration-based calculation are unavailable. Candidate neighborhood sizes from $K = 1$to $K = 15$ were evaluated under the same cross-validation procedure. As shown in Figure 3, the highest classification accuracy of 71.2% was obtained at $K = 11$, which was therefore selected for the final K-NN classifier.

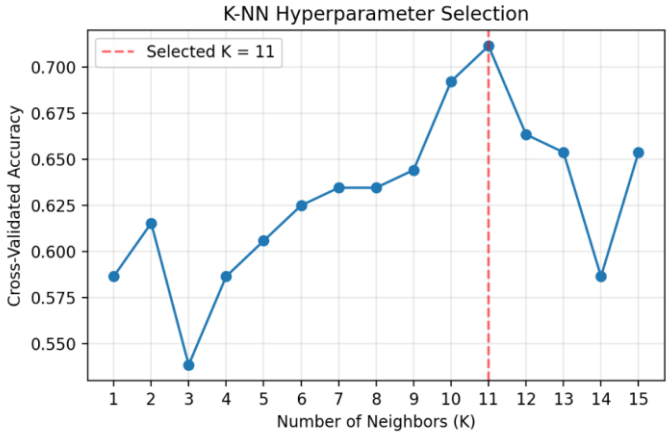

Figure 3: Cross-validated accuracy of the K-NN irrigation-decision classifier as a function of neighborhood size K.

K-NN achieved the highest Accuracy, Precision, weighted F1-score, and Cohen's Kappa among the evaluated classifiers, whereas Logistic Regression obtained the highest

© 2026 The Authors. Journal Compilation © 2026 European Center of Sustainable Development.

ROC-AUC (0.676 compared with 0.667 for K-NN). The slightly higher ROC-AUC of Logistic Regression indicates marginally better overall ranking discrimination across possible classification thresholds; however, the proposed workflow requires a discrete operational decision at the selected operating point rather than ranking performance alone. K-NN was therefore retained because it provided the strongest overall performance at that operating point, including the highest accuracy, weighted F1-score, and agreement beyond chance. Nevertheless, the modest Cohen's Kappa of 0.375 and the class-specific sensitivity results indicate that K-NN should not be regarded as a sufficiently reliable autonomous irrigation controller. Its role in the proposed framework is deliberately limited to an auxiliary decision mechanism when the complete measurements required for evapotranspiration calculation are unavailable, with uncertain outputs subject to confidence-based cloud escalation. Thus, model selection reflects both comparative classification performance and the intended hierarchical operating role rather than ROC-AUC alone.

Table 3: Performance comparison of machine-learning classifiers for the irrigation-decision task using five-fold stratified cross-validation ($n = 104$).

Model	Accuracy	Precision	Recall	F1-Score	Cohen's Kappa	ROC-AUC
K-NN (K = 11)	0.712	0.714	0.712	0.698	0.375	0.667
Random Forest	0.625	0.615	0.625	0.613	0.197	0.644
Logistic Regression	0.635	0.627	0.635	0.61	0.197	0.676
Decision Tree	0.596	0.589	0.596	0.59	0.15	0.598
SVM (RBF)	0.596	0.579	0.596	0.569	0.113	0.42

Figure 4 presents the aggregated confusion matrix for the selected K-NN classifier. Of the 61 no-irrigation observations, 53 were correctly classified, corresponding to a specificity of 86.9%, whereas 21 of the 43 irrigation-required observations were correctly identified, yielding a sensitivity of 48.8%. The model is therefore substantially more reliable in identifying no-irrigation conditions than irrigation-demand events. For this reason, K-NN is used only as an auxiliary decision mechanism: predictions that do not satisfy the predefined 70% confidence criterion are escalated to the cloud-based historical-search mechanism.

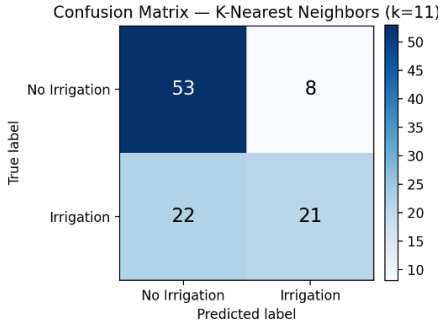

Figure 4: Confusion matrix of the K-NN irrigation-decision classifier (aggregated over 5 folds).

From an operational perspective, false-negative errors are more consequential than false-positive errors because a false negative may incorrectly indicate that irrigation is unnecessary when water is actually required. Repeated false-negative decisions could therefore expose crops to water stress if K-NN were used as an autonomous controller. This asymmetry is one reason the classifier is positioned only as an auxiliary fallback within the proposed hierarchy. The 70% confidence criterion provides a partial safeguard by rejecting low-confidence K-NN outputs and forwarding those cases to the cloud-assisted historical-search stage. However, the threshold does not guarantee elimination of confident but incorrect predictions; consequently, the reported 48.8% sensitivity remains a significant limitation. In practical deployment, the acceptance threshold should therefore be optimized using validation data with greater penalty assigned to false-negative irrigation decisions, and additional agronomic or soil-moisture safety constraints could be incorporated before a no-irrigation prediction is accepted.

5.2 Prototype Functional Validation

The ten-day prototype test verified the integrated operation of sensing, local processing, irrigation-pump actuation, water-flow measurement, data recording, and remote monitoring. Figure 5 illustrates the irrigation experiment conducted during the test period. The prototype successfully acquired sensor measurements, executed irrigation decisions, actuated the pump, recorded delivered water, and transmitted monitoring information, thereby confirming the functional feasibility of the basic sensing-to-actuation workflow. The water-flow measurements were subsequently used in the irrigation-performance analysis presented in Section 5.3.

© 2026 The Authors. Journal Compilation © 2026 European Center of Sustainable Development.

Figure 5: Irrigation process during the prototype evaluation period.

5.3 Irrigation and system performance

The system-level evaluation investigates whether the proposed hierarchical architecture provides benefits beyond basic functional operation. Four aspects are considered: irrigation water consumption, fog-versus-cloud decision latency, resilience to missing sensor information, and communication overhead. Water-consumption performance is evaluated using the recorded sensing dataset, while latency, missing-sensor behavior, and communication overhead are modeled through simulation because the preliminary deployment did not provide the timestamped communication and processing measurements required for direct evaluation. The simulation scenarios were configured to represent the LoRaWAN-enabled fog–cloud architecture defined in Section 3; consequently, the resulting values should be interpreted as modeled performance rather than direct measurements of the physical prototype.

5.3.1 Water-consumption performance

The 52-day sensing dataset was first used to evaluate the irrigation behavior of the proposed adaptive system. During this period, the SA01 site required irrigation on 14 of the 52 monitored days, resulting in a total delivered volume of 7.0 L, whereas SAP01 was irrigated on 29 days with a total volume of 14.5 L. Each recorded irrigation event delivered 0.5 L of water. To provide a reference for comparison, fixed-schedule irrigation was simulated using the same 0.5-L volume per event at intervals ranging from daily irrigation to weekly irrigation.

Table 4: Water consumption: fixed-schedule conventional baseline vs. proposed system (52-day period).

Site	Schedule	Baseline volume (L)	Proposed volume (L)	Water saving
SA01	Every 1 day	26	7	73.10%
SA01	Every 2 days	13	7	46.20%
SA01	Every 3 days	9	7	22.20%

SAP01	Every 1 day	26	14.5	44.20%
SAP01	Every 2 days	13	14.5	-11.50%
SAP01	Every 3 days	9	14.5	-61.10%
Combined	Every 1 day	52	21.5	58.70%
Combined	Every 2 days	26	21.5	17.30%

Table 4 compares the measured water consumption of the proposed system with representative fixed-schedule baselines. Under daily irrigation, the proposed system reduced water consumption from 26.0 to 7.0 L at SA01, corresponding to a 73.1% reduction, and from 26.0 to 14.5 L at SAP01, corresponding to a 44.2% reduction. Across both sites, total consumption decreased from 52.0 L under the daily schedule to 21.5 L using the proposed approach, representing an overall reduction of 58.7%. Relative to an every-two-day schedule, the combined reduction was more modest at 17.3%.

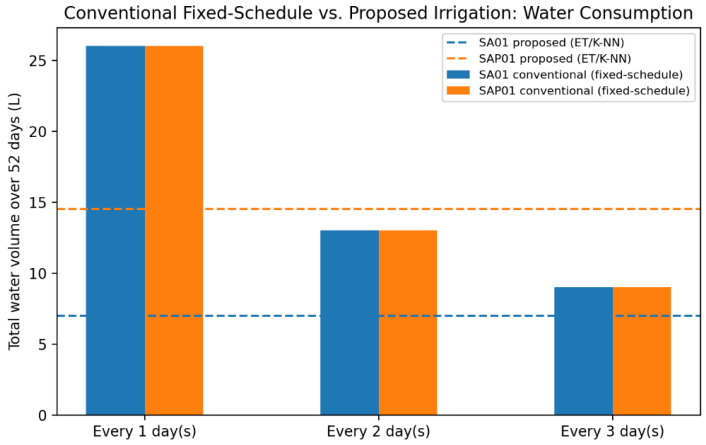

Figure 6: Total water delivered under fixed-schedule baselines (N = 1, 2, 3 days) vs. the proposed adaptive system, per site.

Figure 6 further demonstrates that water-saving percentages depend strongly on the selected reference schedule and site-specific irrigation demand. Although the proposed strategy reduced consumption relative to frequent fixed irrigation, SAP01 received 14.5 L under the adaptive strategy compared with 13.0 L and 9.0 L under the simulated two- and three-day schedules, respectively. These negative "saving" values should not be interpreted as poorer irrigation efficiency in isolation. A schedule can consume less water simply because it supplies less than the crop requires. In contrast, the proposed strategy is intended to respond to measured soil and environmental conditions and estimated crop water demand. Consequently, water efficiency should ultimately be evaluated together with agronomic indicators such as soil-water status, crop stress, yield, and water-use efficiency rather than total delivered volume alone. The differing behavior of SA01 and SAP01 also

© 2026 The Authors. Journal Compilation © 2026 European Center of Sustainable Development.

indicates that local soil, microclimatic, and crop-demand conditions can materially affect the irrigation requirement, supporting zone-specific rather than uniform scheduling.

The magnitude of the observed reduction should also be interpreted in relation to recent smart-irrigation studies. Boukri et al. (2026) reported water savings of approximately 25–35%, Mohamed et al. (2026) reported a 47% reduction, and Rahul et al. (2026) reported a 35.1% reduction. The 58.7% reduction obtained in the present study relative to daily fixed-schedule irrigation is therefore numerically higher than these reported values. However, the percentages are not directly interchangeable because the studies employ different reference irrigation strategies, experimental environments, crops, sensing configurations, and control mechanisms. In particular, the present 58.7% value is calculated against a simulated daily fixed-volume schedule, whereas other studies use different conventional or experimental baselines. The comparison therefore indicates that adaptive irrigation can substantially reduce unnecessary frequent watering, but it should not be interpreted as evidence that the proposed system universally outperforms the reported alternatives. Controlled field comparisons under matched agronomic conditions are required for such a conclusion.

5.3.2 Fog- and cloud-decision latency

Because timestamped end-to-end decision logs were unavailable from the physical deployment, the expected latency of the fog and cloud processing paths was evaluated using a Monte Carlo simulation comprising 5,000 trials. The simulation incorporated sensor-acquisition time, fog-side K-NN inference, actuator-command generation, LoRaWAN transmission, cellular backhaul, cloud processing, and LoRaWAN Class-A downlink behavior. The distributions and assumptions associated with each component are summarized in Table 5.

Table 5: Simulation Parameters for Fog- and Cloud-Decision Latency.

Path segment	Distribution (ms)	Basis
Sensor acquisition (local)	N(50, 12), min 10	Typical microcontroller ADC/bus read
Fog-node K-NN inference	N(35, 10), min 5	Lightweight tabular K-NN on edge hardware
Local actuation dispatch	N(25, 8), min 5	Relay/valve driver command
LoRaWAN uplink airtime	LogNormal(median 400), 80–3000	Simulation distribution parameterized to reflect SF7–SF10 airtime and medium-access/duty-cycle behavior reported in LoRaWAN performance studies (Park et al., 2021; Sørensen et al., 2017).
LoRaWAN Class-A RX1/RX2 wait	U(1000, 2000)	LoRaWAN Class-A receive windows, with RX1 and RX2 opening approximately 1 s and 2 s after uplink transmission, respectively (Park et al., 2021; Sørensen et al., 2017).
Cellular/backhaul (gateway→cloud)	N(150, 40), min 50	Typical fog/cloud backhaul, one-way
Cloud k-d tree search + response	N(120, 35), min 20	Historical-search processing at cloud server

Path segment	Distribution (ms)	Basis

The fog path represents routine decisions processed locally and comprises sensor acquisition, K-NN inference, and actuator-command generation. In contrast, the cloud path includes sensor acquisition, LoRaWAN uplink transmission, cellular backhaul, cloud processing, and the return communication required for actuation. The simulation produced a mean decision latency of approximately 109.7 ms for fog-based processing compared with 2283.7 ms for cloud-based processing. The corresponding 95th-percentile latencies were 138.3 and 2928.5 ms, respectively. Moreover, cloud-path latency exhibited substantially greater variability, with a standard deviation of 386.6 ms compared with 17.3 ms for the fog path.

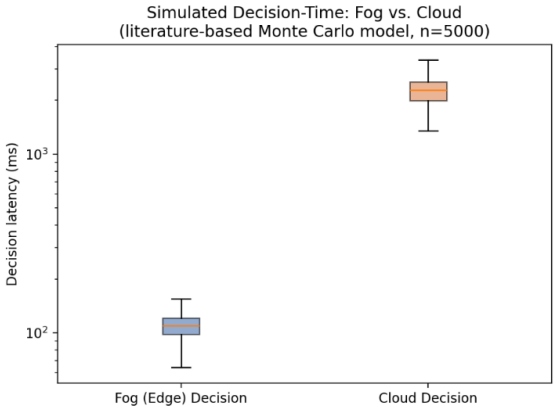

Figure 7: Simulated Decision-Latency Distributions for Fog and Cloud Decision Paths.

As illustrated in Figure 7, the two latency distributions are clearly separated. Based on the mean values, processing routine irrigation decisions at the fog layer yields a modeled latency reduction of approximately 95.2% relative to cloud-based processing. A similar reduction of approximately 95.3% is obtained using the 95th-percentile values.

The simulation further indicates that the principal contributor to cloud-path delay is not the computational complexity of the cloud search itself, but the communication path—particularly the LoRaWAN Class-A receive-window delay. This finding supports the architectural decision to process routine irrigation decisions at the fog layer and invoke cloud processing only when additional historical information is required. Nevertheless, these values represent simulation estimates and should not be interpreted as measured latency of the physical prototype.

5.3.3 Reliability under missing-sensor conditions

The reliability of the hierarchical decision mechanism was further evaluated by introducing controlled levels of missing sensor data into the 104 site-day observations. Random feature missingness ranging from 10% to 50% was introduced, with 300 Monte Carlo trials performed at each missing-data level. Each resulting observation was processed according to the proposed decision hierarchy: complete observations were assigned to the evapotranspiration-based calculation, incomplete observations were processed using the

© 2026 The Authors. Journal Compilation © 2026 European Center of Sustainable Development.

K-NN model ($K = 11$), and K-NN predictions with confidence below the predefined 70% acceptance threshold were escalated to the cloud-based historical-search mechanism.

Table 6: Simulated Decision-Path Distribution Under Sensor-Data Missingness.

Missing rate	ET calculation	K-NN accepted (≥70%)	Cloud fallback
0%	100.00%	0.00%	0.00%
10%	38.80%	21.50%	39.70%
20%	13.80%	30.60%	55.60%
30%	3.90%	34.00%	62.00%
40%	1.00%	34.20%	64.80%
50%	0.20%	32.60%	67.20%

The resulting decision-path distribution is summarized in Table 6 and illustrated in Figure 8. With complete sensor information, all irrigation decisions could be handled by the evapotranspiration-based calculation. At a 10% missing-data rate, however, only 38.8% of decisions remained eligible for direct ET calculation, while 21.5% were resolved by an accepted K-NN prediction and 39.7% required cloud fallback. At 20% missingness, cloud fallback increased to 55.6%, and at missing rates of 30–50%, approximately 62–67% of decisions required cloud assistance.

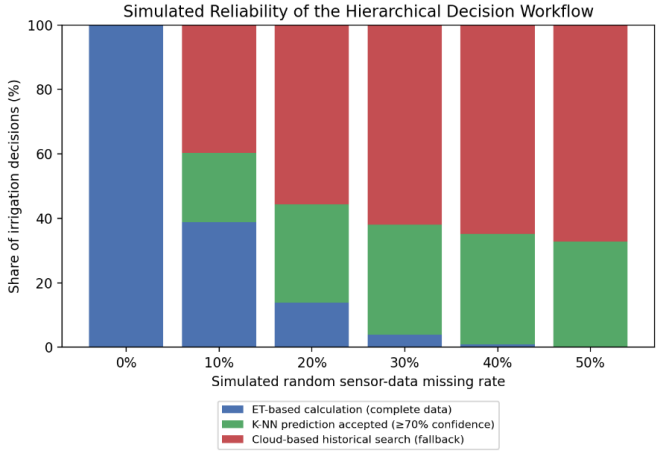

Figure 8: Simulated Distribution of Decision Paths Under Increasing Sensor-Data Missingness.

An important observation is that the increase in cloud fallback is caused not only by missing sensor information but also by the confidence criterion applied to the K-NN output. Predictions below the 70% confidence threshold are deliberately rejected even when the model can technically produce a classification. This behavior indicates that the reliability mechanism operates conservatively, reducing the likelihood that uncertain ML predictions are used directly for irrigation control. At the same time, the results show that

the effectiveness of the fog-first architecture is influenced by the predictive confidence of the local model: a model with improved discrimination and calibration could reduce the frequency of cloud escalation while maintaining the desired reliability level.

5.3.4 Communication overhead

Communication overhead was also evaluated through simulation to examine the potential benefit of resolving routine decisions locally at the fog layer. A 15-min sensing interval was assumed, corresponding to 96 sensor updates per day per node. The modeled communication configuration used a 40-byte LoRaWAN uplink payload, a 20-byte downlink response, and a compact 60-byte hourly summary transmitted to the cloud independently of the irrigation decision path. These quantities represent simulation assumptions rather than measurements obtained from the physical deployment.

Table 7: Simulated Daily Communication Overhead of the Cloud-Centric and Fog-First Architectures.

Missing rate	Baseline msgs/day	Proposed msgs/day	Message reduction	Baseline bytes/day	Proposed bytes/day	Byte reduction
0%	192	24	87.50%	5760	1440	75.00%
10%	192	100.2	47.80%	5760	3725	35.30%
20%	192	130.8	31.90%	5760	4645	19.40%
30%	192	143	25.50%	5760	5011	13.00%

As summarized in Table 7, the cloud-centric baseline requires every sensor update to be transmitted to and processed by the cloud, resulting in 192 messages per day. In contrast, the proposed fog-first architecture resolves routine observations locally and uses wide-area communication primarily for cloud fallback and periodic summary reporting. Under complete sensor availability, the proposed architecture reduces the modeled daily message count from 192 to 24, corresponding to an 87.5% reduction, while the transmitted data volume decreases from 5760 to 1440 bytes per day, corresponding to a 75.0% reduction. As sensor-data missingness increases, the communication advantage gradually decreases because a larger proportion of irrigation decisions are escalated to the cloud. At a 10% missing-data rate, the modeled reductions are 47.8% in message count and 35.3% in transmitted bytes. At 30% missingness, these reductions decrease to 25.5% and 13.0%, respectively.

The communication and reliability results are therefore closely related. Higher missing-data rates increase the use of the cloud fallback mechanism, which in turn increases wide-area traffic and reduces the communication savings obtained from fog-side processing. This relationship highlights an important system-level trade-off: conservative fallback improves decision reliability but increases communication dependency. Improving the robustness and confidence calibration of the local ML model could consequently provide benefits in both reliability and communication efficiency.

5.4 Discussion and Limitations

© 2026 The Authors. Journal Compilation © 2026 European Center of Sustainable Development.

The results collectively demonstrate the complementary roles of the physical, data-driven, and distributed-computing components of the proposed framework. Rather than relying on a single irrigation-decision mechanism, the architecture combines evapotranspiration-based calculation, K-NN prediction, and cloud-assisted historical retrieval according to data availability and prediction reliability. This organization is consistent with the primary design objective of retaining routine decision processing near the field while reserving cloud resources for storage, long-term analysis, and fallback support when local information is insufficient.

The machine-learning results indicate that K-NN is suitable as an auxiliary rather than a standalone irrigation controller. With $K = 11$, the model achieved an accuracy of 71.2% and outperformed the evaluated alternatives in Accuracy, Precision, weighted F1-score, and Cohen's Kappa. However, the confusion-matrix analysis reveals an important asymmetry: the model correctly identified 86.9% of no-irrigation observations but detected only 48.8% of irrigation-required cases. This relatively high false-negative rate limits the extent to which the K-NN output should be trusted independently. In the context of the proposed architecture, however, the model is invoked only when the physically based evapotranspiration calculation cannot be completed, and uncertain predictions can be escalated to the cloud-assisted search mechanism. The ML results therefore support the hierarchical design more strongly than they support the use of K-NN as an autonomous irrigation controller.

The water-consumption analysis further shows that the benefit of adaptive irrigation depends on the baseline against which it is compared. Relative to daily fixed-schedule irrigation, the proposed approach reduced combined water delivery by 58.7%, while the reduction relative to an every-two-day schedule was 17.3%. At SAP01, however, the adaptive system delivered more water than sparse two- or three-day fixed schedules. This finding is important because lower water consumption does not necessarily indicate better irrigation management: a sparse schedule may simply under-irrigate a zone whose environmental and soil conditions require more frequent watering. Accordingly, the objective of the proposed framework should be interpreted as matching irrigation delivery to estimated crop water demand rather than minimizing water volume unconditionally.

The simulation-based latency results support the use of fog-side processing for routine irrigation decisions. The modeled mean latency decreased from approximately 2283.7 ms for the cloud path to 109.7 ms for the fog path, corresponding to a reduction of approximately 95.2%. The difference is primarily associated with the communication path, particularly LoRaWAN Class-A receive-window behavior, rather than with cloud computation alone. This result therefore suggests that the principal advantage of fog processing lies in avoiding unnecessary wide-area round trips for time-sensitive decisions. It should nevertheless be emphasized that these latency values are simulation estimates based on assumed component distributions and are not direct measurements from the deployed prototype.

The missing-sensor experiments reveal a second system-level trade-off between local autonomy and cloud-assisted reliability. At a 10% missing-data rate, 39.7% of decisions required cloud fallback, increasing to 55.6% at 20% missingness and approximately 62–67% for missingness levels between 30% and 50%. The increase is driven not only by unavailable measurements but also by the 70% confidence criterion

applied to the K-NN output. The threshold therefore acts conservatively by preventing uncertain local predictions from being accepted, but this protection comes at the cost of more frequent cloud interaction.

The communication-overhead results exhibit the same dependency. Under complete sensor availability, local processing substantially reduces wide-area communication relative to the cloud-centric baseline. As missingness increases, however, the growing frequency of cloud fallback progressively reduces these savings. The reliability and communication results should therefore not be interpreted independently: stronger local prediction capability or improved confidence calibration could simultaneously reduce unnecessary cloud escalation and lower communication overhead, whereas a highly conservative fallback policy increases communication dependency in exchange for greater protection against uncertain local decisions.

Several limitations should be considered when interpreting these findings. First, the machine-learning evaluation is based on only 104 site-day observations collected from two sensing units during a 52-day period. This limited sample restricts generalization across crops, soil types, seasons, and climatic conditions. The class distribution is also moderately imbalanced, with 43 irrigation-required observations and 61 no-irrigation observations, which contributes to the importance of class-specific evaluation rather than accuracy alone. Although stratified five-fold cross-validation preserves the class distribution across folds, observations from neighboring dates may remain temporally correlated; random site-day partitioning therefore may produce a less stringent estimate of temporal generalization than blocked or forward-in-time validation. Furthermore, the irrigation labels were inferred from recorded water-flow events rather than established through an independent agronomic assessment of crop water need or stress. Together with the modest Cohen's Kappa and the 48.8% irrigation-event sensitivity, these limitations indicate that the ML results should be regarded as preliminary evidence supporting an auxiliary fallback mechanism rather than validation of an autonomous irrigation classifier. Larger multi-season datasets with agronomic ground-truth labels, improved class balance, and time-aware validation are required before broader deployment.

Second, the physical prototype validates the basic sensing, actuation, water-flow measurement, and monitoring workflow, but it does not constitute a full field-scale implementation of all components of the extended fog–cloud architecture. In particular, the current prototype validation does not directly establish the performance of multi-zone LoRa/LoRaWAN communication, large-scale fog deployment, or cloud-based k-d-tree retrieval under real agricultural network conditions. The broader architecture should therefore be regarded as experimentally supported by prototype functionality and simulation-based system analysis rather than as fully field-validated at scale.

Third, the fixed-schedule irrigation baselines are simulated rather than obtained from parallel physical control plots. Consequently, the reported water-saving percentages quantify the difference relative to modeled fixed schedules under the recorded irrigation volume assumptions; they do not yet provide agronomic evidence that the proposed system improves crop yield, plant health, or water-use efficiency relative to a simultaneously operated conventional irrigation treatment. Future field experiments should therefore compare the proposed and conventional strategies under matched crop,

© 2026 The Authors. Journal Compilation © 2026 European Center of Sustainable Development.

soil, and environmental conditions and include agronomic outcomes in addition to total delivered water.

Fourth, the latency and communication results depend on assumed LoRaWAN, backhaul, processing, payload, and sampling parameters. Although the simulation captures the expected behavior of the proposed architecture, actual performance will vary with spreading factor, radio conditions, gateway density, duty-cycle restrictions, cellular connectivity, fog hardware, and cloud-server load. Direct timestamped measurements should therefore be collected in future deployments to validate the modeled latency and communication benefits.

Finally, the predefined 70% acceptance threshold used to determine whether the K-NN prediction is retained locally or escalated to the cloud has not yet been optimized experimentally. The current missing-data results show that this threshold materially affects the balance between local autonomy, reliability, and communication overhead. Future work should investigate threshold calibration using validation data and potentially replace the fixed criterion with a confidence-aware or risk-sensitive policy that accounts explicitly for the higher operational cost of false-negative irrigation decisions.

The results support the proposed hierarchical architecture as a practical framework for combining physically based irrigation estimation, local ML support, and cloud-assisted fallback. The key benefit is not any individual algorithm in isolation, but the ability to select among complementary decision mechanisms according to sensor availability and prediction reliability. At the same time, the present results should be interpreted as an intermediate validation stage: the prototype establishes functional feasibility, the measured dataset supports the irrigation and ML analyses, and the simulations characterize expected system-level behavior, while broader multi-zone field trials remain necessary to establish real-world scalability and agronomic effectiveness.

6. Conclusion and future work

This study presented an intelligent fog–cloud smart irrigation framework that integrates IoT sensing, LoRa/LoRaWAN communication, evapotranspiration-based irrigation calculation, K-NN prediction, and cloud-assisted historical search to support more sustainable and resource-efficient agricultural water management. The proposed hierarchical architecture enables routine irrigation decisions to be processed locally at the fog layer while using the cloud for historical analysis and fallback support when sensor information is incomplete or prediction confidence is insufficient. The K-NN model achieved 71.2% cross-validated accuracy and provided the best overall performance among the evaluated classifiers in several metrics, while prototype testing confirmed the end-to-end operation of sensing, actuation, water-flow monitoring, and remote visualization. Using the recorded 52-day dataset, the adaptive strategy reduced combined water consumption by 58.7% relative to a simulated daily fixed-schedule baseline and by 17.3% relative to an every-two-day schedule. Simulation results further indicated a 95.2% reduction in mean decision latency for fog processing and communication reductions of up to 87.5% in message count and 75.0% in transmitted data volume under complete sensor availability. These findings demonstrate the potential of combining physically based irrigation estimation, local machine-learning support, and cloud-assisted fallback within a unified framework. Future research will prioritize five complementary directions. First,

longer-term multi-zone field trials across different crops, soils, seasons, and climatic conditions will be conducted to establish agronomic effectiveness and system generalizability. Second, larger and more balanced datasets with independent crop-water-demand or crop-stress ground truth will be collected to improve irrigation-event detection and support time-aware model validation. Third, the K-NN acceptance threshold will be optimized using validation data and risk-sensitive criteria that assign greater cost to false-negative irrigation decisions. Fourth, the proposed architecture will be evaluated using direct timestamped measurements of LoRaWAN transmission, fog processing, cloud communication, and energy consumption rather than simulation alone. Finally, field-scale experiments will compare the adaptive strategy with simultaneously operated conventional irrigation controls using crop yield, plant health, soil-water status, and water-use efficiency in addition to delivered water volume. These evaluations are necessary to establish the scalability, agronomic effectiveness, and practical sustainability benefits of the proposed framework under realistic agricultural conditions.

Acknowledgment: This work was funded by the University of Jeddah, Jeddah, Saudi Arabia, under grant No. (UJ-23-FR-44). Therefore, the authors thank the University of Jeddah for its technical and financial support

References

Allen, R. G., Pereira, L. S., Raes, D., & Smith, M. (1998). *Crop evapotranspiration: Guidelines for computing crop water requirements* (FAO Irrigation and Drainage Paper No. 56). Food and Agriculture Organization of the United Nations.

Ayat, Y., El Moussati, A., Rachdi, O., Dinar, M., El Aouni, A., Karkri, H., Benzaouia, M., Benzekri, W., Mir, I., El Amrani, A., & Mimouni, A. (2026). Design and experimental validation of a low-power IoT-based smart irrigation system using LoRa, ET0, and Crop Water Stress Index for precision agriculture. *IoT, 7*(3), Article 59. https://doi.org/10.3390/iot7030059

Balfaqih, M. (2023). Design and implementation of IoT-enabled intelligent irrigation system using machine learning. In *2023 28th Asia Pacific Conference on Communications (APCC)* (pp. 436–441). IEEE.

Bautista, C. J. C., Gallegos, R. K. B., & Lampayan, R. M. (2026). Machine learning and digital twins in smart irrigation: Optimising water use through agricultural data analytics. *Digital Twin, 3*(1), 2562418. https://doi.org/10.1080/27525783.2025.2562418

Boukri, Y., Hamici, H. S., Mansour, R. F., Maamar, A. E. T., Ghoneim, S. S. M., Paramasivam, P., Hashim, M. A., & Hussein, E. E. (2026). Analysis and experimental implementation of affordable smart irrigation system using IoT to reduce agricultural costs and minimize water usage. *Applied Water Science, 16*, Article 36. https://doi.org/10.1007/s13201-025-02727-4

Brouwer, C., & Heibloem, M. (1986). *Irrigation water management: Irrigation water needs* (Training Manual No. 3). Food and Agriculture Organization of the United Nations.

Calera, A., Campos, I., Osann, A., D'Urso, G., & Menenti, M. (2017). Remote sensing for crop water management: From ET modelling to services for the end users. *Sensors, 17*(5), 1104.

Chen, X., Dhungel, J., Bhattarai, S. P., Torabi, M., Pendergast, L., & Midmore, D. J. (2011). Impact of oxygation on soil respiration, yield and water use efficiency of three crop species. *Journal of Plant Ecology, 4*(4), 236–248.

Dutta, R., Morshed, A., Aryal, J., D'Este, C., & Das, A. (2014). Development of an intelligent environmental knowledge system for sustainable agricultural decision support. *Environmental Modelling & Software, 52*, 264–272.

Goap, A., Sharma, D., Shukla, A. K., & Krishna, C. R. (2018). An IoT based smart irrigation management system using machine learning and open source technologies. *Computers and Electronics in Agriculture, 155*, 41–49.

© 2026 The Authors. Journal Compilation © 2026 European Center of Sustainable Development.

Gonzales-Gustavson, E., Rusiñol, M., Medema, G., Calvo, M., & Girones, R. (2019). Quantitative risk assessment of norovirus and adenovirus for the use of reclaimed water to irrigate lettuce in Catalonia. *Water Research, 153*, 91–99.

Gutiérrez, J., Villa-Medina, J. F., Nieto-Garibay, A., & Porta-Gándara, M. Á. (2014). Automated irrigation system using a wireless sensor network and GPRS module. *IEEE Transactions on Instrumentation and Measurement, 63*(1), 166–176.

Kashyap, P. K., Kumar, S., Jaiswal, A., Prasad, M., & Gandomi, A. H. (2021). Towards precision agriculture: IoT-enabled intelligent irrigation systems using deep learning neural network. *IEEE Sensors Journal, 21*(16), 17479–17491.

Liu, Y., Yan, H., Zhang, C., Zhang, J., Zhang, D., Bao, R., & Han, Y. (2026). Application of irrigation decision-making methods for smart irrigation systems: A review. *Irrigation and Drainage, 75*(3), 1250–1264. https://doi.org/10.1002/ird.70098

Martínez-Alvarez, V., Martin-Gorriz, B., & Soto-García, M. (2016). Seawater desalination for crop irrigation—A review of current experiences and revealed key issues. *Desalination, 381*, 58–70.

Mehlhorn, K. (2013). *Data structures and algorithms 1: Sorting and searching* (Vol. 1). Springer Science & Business Media.

Mohamed, Z. E., Afify, M. K., Badr, M. M., & Omar, O. A. (2026). IoT-driven smart irrigation system to improve water use efficiency. *Scientific Reports, 16*, Article 2609. https://doi.org/10.1038/s41598-025-33826-6

Mohammed, M. E. A., Alhajhoj, M. R., Ali-Dinar, H. M., & Munir, M. (2020). Impact of a novel water-saving subsurface irrigation system on water productivity, photosynthetic characteristics, yield, and fruit quality of date palm under arid conditions. *Agronomy, 10*(9), 1265.

Morchid, A., Qjidaa, H., El Alami, R., Mobayen, S., Skruch, P., & Bossoufi, B. (2026). Smart irrigation-based internet of things and cloud computing technologies for sustainable farming. *Scientific Reports, 16*, Article 5293. https://doi.org/10.1038/s41598-026-35810-0

Park, J., Kim, K., & Choi, J. P. (2021). An implementation design of unified protocol architecture for physical layer of LoRaWAN end-nodes. *Electronics, 10*(20), 2550. https://doi.org/10.3390/electronics10202550

Rahul, S. G., Rithikkaa, S. J., Amirthalakshmi, T. M., & Gayathri, M. (2026). Sustainable agriculture through IoT-driven smart irrigation with explainable AI. *Frontiers in Sustainable Food Systems, 10*, 1847041. https://doi.org/10.3389/fsufs.2026.1847041

Rani, N. S., & Banu, N. S. (2026). Cognitive integration of internet of things and feedforward learning models for smart irrigation in sustainable agriculture. *Scientific Reports, 16*, Article 17413. https://doi.org/10.1038/s41598-025-29497-y

Salvi, S., Jain, S. F., Sanjay, H. A., Harshita, T. K., Farhana, M., Jain, N., & Suhas, M. V. (2017). Cloud based data analysis and monitoring of smart multi-level irrigation system using IoT. In *2017 International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC)* (pp. 752–757). IEEE.

Saqib, M., Almohamad, T. A., & Mehmood, R. M. (2020). A low-cost information monitoring system for smart farming applications. *Sensors, 20*(8), 2367.

Shrivastava, P., & Kumar, R. (2015). Soil salinity: A serious environmental issue and plant growth-promoting bacteria as one of the tools for its alleviation. *Saudi Journal of Biological Sciences, 22*(2), 123–131.

Sørensen, R. B., Kim, D. M., Nielsen, J. J., & Popovski, P. (2017). Analysis of latency and MAC-layer performance for Class A LoRaWAN. *IEEE Wireless Communications Letters, 6*(5), 566–569.

Torres-Sánchez, R., Navarro-Hellín, H., Guillamón-Frutos, A., San-Segundo, R., Ruiz-Abellón, M. C., & Domingo-Miguel, R. (2020). A decision support system for irrigation management: Analysis and implementation of different learning techniques. *Water, 12*(2), 548.

Tzounis, A., Katsoulas, N., Bartzanas, T., & Kittas, C. (2017). Internet of Things in agriculture, recent advances and future challenges. *Biosystems Engineering, 164*, 31–48.

Zhang, N., Wang, M., & Wang, N. (2002). Precision agriculture—A worldwide overview. *Computers and Electronics in Agriculture, 36*(2–3), 113–132.