A Deployable Digital Twin System for Precision Irrigation Management via Deep Reinforcement Learning

Jiamei Liu, Fangle Chang, Longhua Ma, and Hongye Su

Abstract—**Deep reinforcement learning (DRL) shows great potential in irrigation scheduling. However, deploying complex DRL models on field hardware remains difficult. This paper proposes a deployable smart irrigation system. We integrate a biophysical crop growth model with an agent. A modular Python-Flask backend enables efficient real-time inference. We decoupled inference from training to reduce computational overhead. The system uses a bidirectional long short-term memory encoder to extract temporal features. We developed a browser/server (B/S) architecture dashboard for real-time monitoring and explainable decision heatmaps. System-level validation was conducted using historical data from Xinjiang cotton fields. The results show sub-10 ms inference latency and a 0.45 s total system response time. The system provides a 1.9% yield and 5.9% irrigation water use efficiency (IWUE) boost over standard soft actor-critic (SAC) algorithms, proving its algorithmic efficacy. This architecture transforms theoretical gains into executable irrigation schedules for real-world farming. It ensures stable and failure-free operation across continuous growing seasons.**

I. Introduction

Water scarcity constrains sustainable agriculture globally, especially in arid, irrigation-dependent regions [1]. Due to prevalent empirical or fixed-schedule practices, irrigation efficiency remains suboptimal [2, 3]. Such static management fails to adapt to the non-linear dynamics of the soil plant atmosphere continuum (SPAC), leading to water and yield loss from untimely stress [4]. Therefore, data-driven optimization of spatiotemporal water allocation is needed [5].

Irrigation technology has evolved significantly. Early automation relied on simple, low-adaptability timers [6]. The Internet of Things (IoT) later enabled real-time monitoring, using sensor networks and cloud platforms (e.g., ThingsBoard) to visualize soil and weather data [7–9]. However, most IoT solutions still rely on basic threshold controls and lack the predictive capability for complex environments [10, 11]. To address this, artificial intelligence (AI), specifically deep reinforcement learning (DRL), has emerged for cognitive control [12]. DRL agents learn optimal strategies through trial-and-error, theoretically outperforming heuristic rules by maximizing long-term rewards [13–15].

Despite DRL's algorithmic success in simulation, a deployment gap remains between theory and engineering reality [16]. Most studies are confined to numerical experiments, lacking the software architecture to connect policy networks with real-world data pipelines [17, 18]. DRL models are often computationally intensive, creating challenges for deployment on resource-constrained edge nodes due to memory limitations [19]. Additionally, farmers require not just a decision value but also an explanation of the underlying logic, which is often missing in existing systems [20]. To bridge this gap, digital twins (DTs) offer a viable framework [21]. In irrigation, a digital twin can simulate soil water balance and crop development driven by real-time weather data [22, 23]. By integrating biophysical models with AI agents, it enables risk-free decision support [24].

This paper presents the design of a deployable AI agent for precision irrigation. The main contributions are summarized as follows:

(1) We developed a decoupled Python-Flask backend that encapsulates the Fortran-based decision support system for agrotechnology transfer (DSSAT) engine and the PyTorch-based DRL agent.

(2) We deployed an optimized agent with bidirectional long short-term memory (BiLSTM) and attention mechanisms. Unlike standard DRL applications that assume Markovian independence, our agent addresses the phenological time-lag effect by reformulating the state space. While existing DRL models suffer from policy noise in stochastic weather, our dual-attention mechanism forces the agent to prioritize critical biophysical stressors over transient environmental fluctuations.

(3) We conducted system-level validation using historical data from Xinjiang cotton fields. The system achieved an 8.4% increase in seed cotton yield while reducing total irrigation volume by 22.1% compared to the traditional methods. Furthermore, the proposed architecture outperformed the standard soft actor-critic (SAC) baseline by 1.9% in yield and 5.9% in irrigation water use efficiency (IWUE).

Manuscript received: 10 February 2026; revised: 6 March 2026; accepted: 16 March 2026. *(Corresponding author: Fangle Chang.)*

Citation: J. Liu, F. Chang, L. Ma, and H. Su, A deployable digital twin system for precision irrigation management via deep reinforcement learning, *Int. J. Intell. Control Syst.*, 2026, 31(1), 100–105.

Jiamei Liu and Fangle Chang are with Ningbo Global Innovation Center, Zhejiang University, Ningbo 315100, China (e-mail: liujiam06@163.com; fanglechang@zju.edu.cn).

Longhua Ma is with School of Information Science and Engineering, NingboTech University, Ningbo 315100, China (e-mail: lhma_zju@zju.edu.cn).

Hongye Su is with College of Control Science and Engineering, Zhejiang University, Hangzhou 310058, China (e-mail: hysu69@zju.edu.cn).

Digital Object Identifier 10.62678/IJICS202603.10296

II. Proposed System

A. System Overview

To overcome the limitations of traditional hardware controllers relying on static thresholds, this paper proposes an AI system for smart irrigation. The system integrates a high-fidelity crop simulation engine, a deep reinforcement learning agent, and a visualization platform into a unified software framework. By accessing real-time weather data, the system dynamically reconstructs the soil-plant-atmosphere continuum in a virtual environment, predicts water requirements, and autonomously outputs precise irrigation schedules. Designed to be hardware-agnostic and energy-efficient, the system can run on edge gateways, transforming high-dimensional environmental data into executable irrigation commands in real-time.

B. System Architecture

The system architecture is designed as a modular three-tier framework, as shown in Fig. 1.

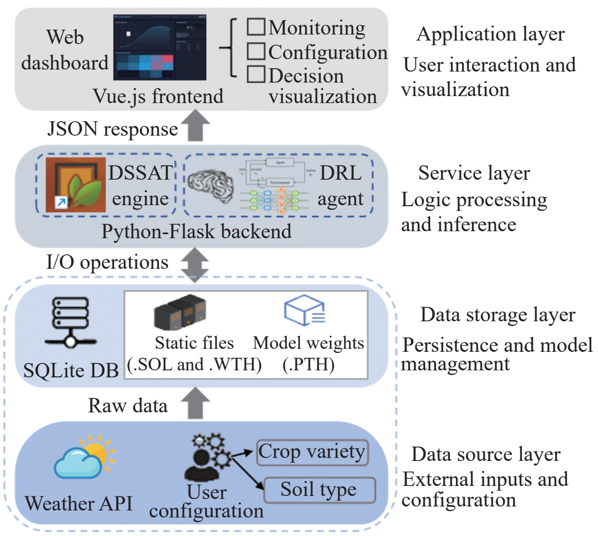

Fig. 1 Overall architecture of the proposed system for smart irrigation.

Data layer: This layer serves as the infrastructure foundation, responsible for acquiring and managing agricultural data. It integrates external inputs, such as real-time meteorological data fetched from weather bureaus and user-defined field configurations. For persistence, an SQLite database is used to log historical decision records, while the file system stores DSSAT configuration files (.SOL and .WTH) and pre-trained DRL model weights (.PTH). This layer provides the essential data inputs for the simulation engine.

Service layer: Acting as the core processing unit, this layer is built on Python-Flask. It encapsulates two key engines:

(1) Simulation engine: Wraps the DSSAT v4.8 kernel to simulate crop growth dynamics and soil water balance on a daily time-step.

(2) Inference engine: Hosts the pre-trained DRL agent using PyTorch. It receives the simulated state vector from the DSSAT engine and computes the optimal irrigation action.

Application layer: Developed with Vue.js and ECharts, this layer provides a responsive web interface. It features a real-time monitoring dashboard, a system configuration panel, and interpretable visualizations, enabling farmers to understand the rationale behind the AI's decisions.

C. Software Specifications

Unlike hardware-centric IoT systems, the performance and scalability of our proposed solution rely heavily on an optimized software stack. The key software components and their specific versions used in the implementation are detailed in Table 1.

Table 1 Software components and specifications.

Component	Technology stack	Functionality
Backend framework	Python 3.8/Flask 2.0	API handling and process scheduling
AI framework	PyTorch 1.10 (CUDA 11.3)	DRL model loading and inference
Simulation kernel	DSSAT v4.8 (CSM-CROPGRO)	Biophysical crop modeling
Frontend framework	Vue.js 2.6/Element UI	User interface and interaction
Visualization	ECharts 5.0	Dynamic charting of crop status
Database	SQLite 3.35	Lightweight data persistence

D. Intelligent Control Algorithm

We model the irrigation scheduling problem as a partially observable Markov decision process (POMDP) and solve it using an enhanced soft actor-critic algorithm integrated with temporal-spatial attention mechanisms.

Unlike methods that rely on instantaneous sensor readings, crop growth is a cumulative process with significant time-lag effects. To capture these dynamics, we construct a sliding window of historical observations. Let o_t denote the observation vector at day t, comprising the air temperature T, the precipitation amount P, the solar radiation SR, the soil water content SW, and crop biomass B. The input state S_t is defined as

$$S_t = \{o_{t-L+1}, o_{t-L+2}, ..., o_t\} \tag{1}$$

where $L = 7$ is the time window size. This value was determined through a sensitivity analysis balancing predictive accuracy and computational overhead. Compared to a shorter window ($L = 3$), the $L = 7$ configuration improved the yield prediction accuracy by 4.3% by capturing the cumulative effects of moisture stress. While extending the window to $L = 14$ provided a marginal gain in precision (0.6%), it increased the inference latency on the edge gateway by 140%. Thus, $L = 7$ was selected as the optimal threshold for capturing the weekly meteorological cycle while ensuring real-time performance on resource-constrained hardware. This sequence is fed into a BiLSTM encoder to extract a hidden feature representation h_t that encapsulates temporal dependencies.

A dual-attention module is applied to h_t. Temporal attention assigns weights α_i to different time steps to highlight critical events (e.g., heavy rainfall yesterday). Feature attention assigns weights β_j to specific environmental factors (e.g., distinguishing between moisture stress and temperature stress). The final context vector C_t is a weighted sum of features, serving as the input for the policy network.

The decision-making core employs the SAC framework, an off-policy algorithm that optimizes a maximum entropy objective. This approach encourages the agent to explore the state space while maximizing the expected reward, preventing convergence to sub-optimal policies in the complex agricultural environment. The objective function $\mathcal{J}(\pi)$ is defined as

$$\mathcal{J}(\pi) = \sum_{t=0}^{T} E_{(S_t, a_t) \sim \rho_\pi} \left[r(S_t, a_t) + \alpha \mathcal{H} \pi(\cdot | S_t) \right] \tag{2}$$

where $r(S_t, a_t)$ denotes the immediate reward received at state S_t and action a_t. $\mathcal{H}(\cdot)$ represents the entropy of the policy π, quantifying the randomness of the agent's actions. α is the temperature parameter that balances the trade-off between reward maximization and entropy maximization. ρ_π is the state-action marginal distribution induced by the policy.

The SAC agent consists of an actor network and two critic networks working in tandem to optimize the irrigation policy. The actor network is responsible for mapping the context vector to a probability distribution over actions. It outputs the mean μ and standard deviation σ of a Gaussian distribution. To enable backpropagation through the stochastic sampling process, the reparameterization trick is applied. The critic networks, on the other hand, consist of two independent Q-networks (Q_1 and Q_2) that estimate the soft action-value function $Q(S_t, a_t)$. During the training phase, the minimum Q-value from these two networks is utilized for the Bellman update calculation to mitigate the overestimation bias in value-based reinforcement learning methods.

To align the agent's behavior with agronomic principles, we design a stage-dependent reward function R_t. This function balances the trade-off between yield maximization and water conservation at each time step. The reward structure is formulated as

$$R_t = \mathcal{W}_1 \Delta Y - \mathcal{W}_2 I - \mathcal{W}_3(G) \mathcal{N}^2 \tag{3}$$

where ΔY represents the incremental gain in crop biomass or yield potential compared to the previous day. I denotes the amount of irrigation water applied on day t. $\mathcal{N} \in [0,\ 1]$ is the soil water stress factor simulated by DSSAT, where 1 indicates severe stress and 0 indicates no stress. We employ a quadratic penalty term to disproportionately punish severe stress conditions, thereby encouraging the agent to maintain soil moisture within a safe range. $\mathcal{W}_1$ and $\mathcal{W}_2$ are fixed weighting coefficients for yield gain and water cost, respectively. $\mathcal{W}_3(G)$ is a dynamic weight coefficient that varies according to the crop's growth stage ID (GSTD).

The dynamic nature of $\mathcal{W}_3(G)$ is quantified according to the sensitivity of Xinjiang cotton at different growth stages. Specifically, $\mathcal{W}_3$ is set to 5.0 during the seedling and vegetative stages. During the flowering and boll-setting stage (GSTD 3–5), $\mathcal{W}_3$ is increased to its maximum value of 20.0 to ensure that the soil water stress factor $\mathcal{N}$ remains near zero. In the boll opening stage (GSTD 6), $\mathcal{W}_3$ is reduced to 2.0, allowing the agent to tolerate a mild soil water deficit to promote fiber maturation.

The complete execution flow of the proposed system is summarized in Algorithm 1.

Algorithm 1 SAC driven intelligent irrigation control loop

```
Initialize reply memory 𝒟 and state queue Q_L,
initialize training environment env = DSSAT(·), and
randomly initialize actor network π_θ and critic networks
Q_φ1 and Q_φ2;
For episode from 1 to M do
    s_0 = env.reset(·);
    Push s_0 into Q_L to form initial sequence S_0;
    done = False;
    t = 0;
    While done = False and t < T_max do
        Get observation sequence S_t from Q_L;
        Extract features h_t = BiLSTM(S_t);
        Compute context C_t = Attention(h_t);
        Sample action a_t ~ π_θ(·|C_t);
        Clip action a_t ∈ [0, 60];
        s_{t+1}, r_t, done = env.step(a_t);
        Push s_{t+1} into Q_L to update S_{t+1};
        Store tuple (S_t, a_t, r_t, S_{t+1}) in 𝒟;
        if |𝒟| > BatchSize then
            sample random mini-batch from 𝒟;
            update critic network parameter φ by
              minimizing Bellman error;
            update actor network parameter θ by
              maximizing objective 𝒥(π);
            soft-update target networks;
        end
        t = t + 1;
    end
end
```

III. System Implementation

A. Backend Implementation

The system backend is developed using Python 3.8 and the Flask micro-framework. To enable interaction with the legacy Fortran-based DSSAT engine, we implemented a custom wrapper module using the subprocess library. This wrapper handles the dynamic generation of .INP experiment files and parses the binary .OUT results in real-time. To ensure high concurrency, a file-locking mechanism is introduced to prevent read/write conflicts during simulation steps. The DRL inference engine is built on PyTorch. Upon system startup, the pre-trained SAC model weights (.PTH) are loaded into the graphics processing unit (GPU) memory, ensuring that the inference latency for each decision step remains below 10 ms. The agent was trained for 2000 episodes, where each episode represents a full growing season (approximate 150 days).

The actor and critic networks both consist of two hidden layers with 256 neurons each and rectified linear unit (ReLU) activation functions. To ensure stability, we utilized a replay buffer of 10^5 transitions, a batch size of 256, and a discount factor of 0.99. Both networks were optimized using the Adam optimizer with a learning rate of 3×10^{-4}. The target network update rate was set to $\tau = 0.005$. Training was conducted over five independent random seeds to evaluate statistical stability, taking approximately 5.5 h on an NVIDIA RTX 3060 GPU. Policy convergence was consistently observed around 1200 episodes, with no significant policy collapse or performance variance.

B. Frontend and Interaction Design

The user interface is constructed as a single page application using Vue.js. As shown in Fig. 2, the system configuration panel allows users to initialize the digital twin environment by selecting specific crop cultivars (e.g., Xinluzao-series cotton) and soil profiles. This modular design ensures the system's adaptability to different agricultural regions without code modification.

C. Visualization and Explainability

To bridge the gap between complex algorithms and end-users, we integrated ECharts for high-performance data visualization. The main dashboard (Fig. 3) features a real-time crop monitor, which renders the simulated growth trajectories (leaf area index, LAI) synchronized with daily weather updates. A unique decision explainability module visualizes the attention weights assigned by the SAC agent to different environmental factors, providing farmers with transparent and interpretable decision support.

IV. Experimental Evaluation

A. Experimental Setup

The experimental validation was conducted using data from a cotton production region in Shihezi, Xinjiang (44°18′ N, 85°59′ E). To drive the digital twin environment, we utilized historical meteorological records spanning from 2022 to 2024. To rigorously evaluate the performance of the proposed SAC agent, two traditional irrigation strategies were established as baselines: (1) fixed-schedule irrigation (FSI), which applies a fixed quota of 45 mm every 10 days, reflecting local empirical practices; and (2) soil moisture threshold (SMT), a reactive rule-based method that triggers irrigation whenever the root-zone soil water content drops below 60% of the field capacity.

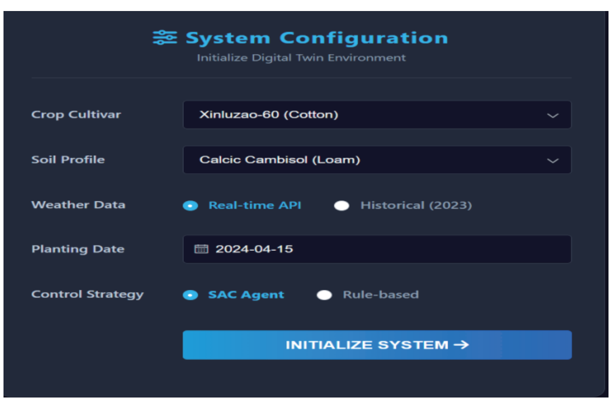

Fig. 2 System configuration panel.

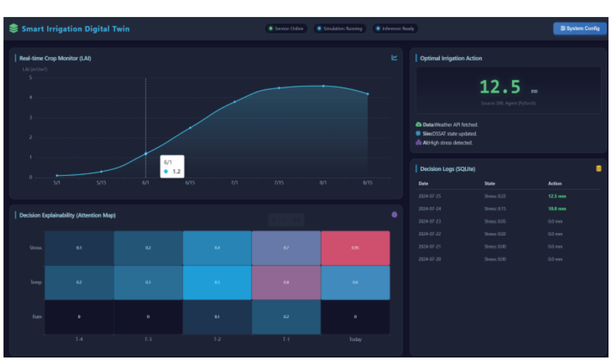

Fig. 3 Main user interface of the intelligent irrigation system.

B. Evaluation of Digital Twin Fidelity

Prior to deploying the intelligent agent, the fidelity of the DSSAT-based digital twin environment was rigorously validated against field data. Fig. 4 presents the scatter plots comparing the simulated versus observed values for above-ground biomass and seed cotton yield. Quantitative analysis reveals that the model achieves a high coefficient of determination ($R^2 = 0.78$ for biomass and $R^2 = 0.82$ for yield), with a normalized root mean square error (nRMSE) of 9.5% for biomass and 7.2% for yield across both calibration (2022) and validation (2023 to 2024) datasets.

As observed in Fig. 4, the 2022 calibration points exhibit a slight underestimation, which is attributed to the extreme heat stress recorded during the July 2022 flowering stage. Under these conditions, the DSSAT model's physiological subroutines calculated more severe leaf senescence and reduced carbon assimilation than what was measured *in situ*. In contrast, the more moderate weather of 2023 and 2024 resulted in a more balanced distribution of validation points around the 1:1 line. Despite the 2022 bias, the overall simulation error remains within the acceptable precision range for driving the DRL agent.

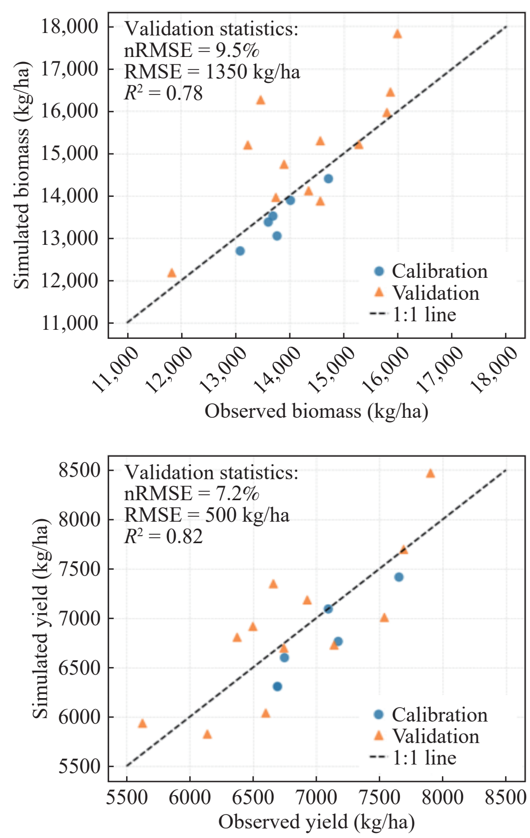

Fig. 4 Comparison of simulated value and observed value for above-ground biomass and seed cotton yield during calibration (2022) and validation (2023 to 2024) periods. The dashed line represents the 1:1 relationship.

C. Irrigation Performance Analysis

Table 2 summarizes the comparative performance metrics. The results indicate that the agent achieved a seed cotton yield of 7421 kg/ha, representing an 8.4% increase over the FSI baseline. Due to the synergistic effect of yield enhancement and water reduction, the IWUE exhibited a substantial improvement of 39.0%. Crucially, when compared with the standard SAC algorithm, our agent achieved an additional 1.9% yield gain and 5.9% IWUE improvement. This marginal yet significant gain demonstrates the methodological superiority of our attention-augmented architecture in capturing complex crop-water dynamics over vanilla DRL methods.

Table 2 Performance comparison of different irrigation strategies. Bold values indicate the best performance among all evaluated strategies.

Irrigation strategy	Yield (kg/ha)	Irrigation (mm)	IWUE (kg/(ha·mm))
FSI	6845	485	14.1
SMT	7120	410	17.4
Proximal policy optimization (PPO)	7215	395	18.2
SAC	7280	392	18.5
Agent	**7421**	**378**	**19.6**

Fig. 5 illustrates the training convergence, where the proposed agent achieves faster learning and a significantly higher reward ceiling than PPO and standard SAC baselines. The reduced performance variance in the later stages confirms the superior stability of our attention-augmented architecture in stochastic environments.

Ablation studies (Table 2 and Fig. 5) demonstrate the necessity of the BiLSTM-attention architecture. Fig. 5 confirms that without the attention module, standard DRL baselines exhibit intense fluctuations and slower convergence in stochastic DSSAT environments.

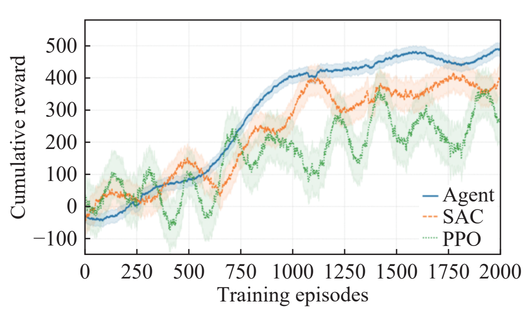

Fig. 5 Comparison of training convergence and stability.

D. System Latency and Stability

To validate the practical deployability, the system was evaluated on a resource-constrained edge gateway featuring a quad-core ARM processor (1.5 GHz) and 4 GB of RAM running on Ubuntu 20.04. The average end-to-end latency for a single daily decision cycle is 0.45 s, which includes 150 ms for data acquisition, 280 ms for DSSAT simulation, and less than 10 ms for agent inference. This low computational overhead ensures that the system can provide real-time responses even on low-power hardware. Furthermore, the system demonstrated high reliability during a 1000-day continuous simulation test, maintaining a stable memory footprint of approximately 180 MB without any runtime exceptions or crashes.

V. Conclusion

This paper presented a deployable intelligent irrigation system that bridges the gap between sophisticated deep reinforcement learning algorithms and practical agricultural engineering. By integrating a high-fidelity DSSAT-based digital twin with an SAC agent, we developed a framework capable of capturing complex crop-water dynamics and executing optimal irrigation strategies. The system architecture, built on a decoupled Python-Flask and Vue.js stack, ensures computational efficiency while providing interpretable decision support. The verified robustness across multi-year Xinjiang datasets suggests that the agent framework can bridge the deployment gap in smart agriculture.

Experimental evaluations conducted using historical data from Xinjiang demonstrate the advantages of the proposed architecture, which outperformed the standard SAC baseline by 1.9% in yield and 5.9% in IWUE, validating the effectiveness of the attention-based BiLSTM encoder.

Future work will focus on two primary directions. First, we aim to extend the system's generalizability by training and validating the agent across diverse crop varieties and different climatic regions. Second, we plan to incorporate multi-agent reinforcement learning (MARL) to optimize the collaborative water allocation of multiple irrigation units at a regional scale.

Acknowledgment

This work was supported by the National Natural Science Foundation of China (No. 62476243).

References

[1] Y. Jiao, G. Zhu, G. Meng, S. Lu, D. Qiu, X. Lin, R. Li, Q. Wang, L. Chen, L. Zhao et al., Estimating non-productive water loss in irrigated farmland in arid oasis regions: Based on stable isotope data, *Agric. Water Manag.*, 2023, 289, 108515.

[2] J. P. Padilla-Nates, L. D. Garcia, C. Lozoya, L. Orona, and A. Cortes-Perez, Greenhouse irrigation control based on reinforcement learning, *Agronomy*, 2025, 15(12), 2781.

[3] Ž. Kavaliauskas, G. Blažiūnas, and I. Šajev, Optimization of an automated substrate irrigation system using the SAC reinforcement learning agent, *Appl. Sci.*, 2025, 15(23), 12715.

[4] Y. Zeng, A. Verhoef, H. Vereecken, E. Ben-Dor, T. Veldkamp, L. Shaw, M. Van Der Ploeg, Y. Wang, and Z. Su, Monitoring and modeling the soil-plant system toward understanding soil health, *Rev. Geophys.*, 2025, 63(1), e2024RG000836.

[5] H. Zhao, L. Di, L. Guo, C. Zhang, and L. Lin, An automated data-driven irrigation scheduling approach using model simulated soil moisture and evapotranspiration, *Sustainability*, 2023, 15(17), 12908.

[6] Y. Liu, H. Yan, C. Zhang, J. Zhang, G. Wang, D. Zhang, R. Bao, and Y. Han, Application of irrigation decision-making methods for smart irrigation systems: A review, *Irrig. Drain.*, to be published.

[7] A. Morchid, R. Jebabra, H. M. Khalid, R. El Alami, H. Qjidaa, and M. O. Jamil, IoT-based smart irrigation management system to enhance agricultural water security using embedded systems, telemetry data, and cloud computing, *Results Eng.*, 2024, 23, 102829.

[8] B. Et-Taibi, M. R. Abid, E. M. Boufounas, A. Morchid, S. Bourhnane, T. A. Hamed, and D. Benhaddou, Enhancing water management in

smart agriculture: A cloud and IoT-based smart irrigation system, *Results Eng.*, 2024, 22, 102283.

[9] A. Hasib and A. Akib, An IoT-based smart plant monitoring and irrigation system with real-time environmental sensing, automated alerts, and cloud analytics, arXiv preprint arXiv: 2601.15830, 2026.

[10] M. Dhanaraju, P. Chenniappan, K. Ramalingam, S. Pazhanivelan, and R. Kaliaperumal, Smart farming: Internet of Things (IoT)-based sustainable agriculture, *Agriculture*, 2022, 12(10), 1745.

[11] S. Mansoor, S. Iqbal, S. M. Popescu, S. L. Kim, Y. S. Chung, and J. H. Baek, Integration of smart sensors and IOT in precision agriculture: Trends, challenges and future prospectives, *Front. Plant Sci.*, 2025, 16, 1587869.

[12] J. Liu, F. Chang, J. Yang, X. Jie, C. Lu, C. Wang, L. Xie, L. Ma, and H. Su, Deep reinforcement learning for irrigation optimization: Advantages, opportunities, and challenges, *Agric. Water Manag.*, 2025, 322, 110030.

[13] J. Liu, F. Chang, X. Wang, M. Kang, C. Lu, C. Wang, S. Hu, Y. Li, L. Ma, and H. Su, Smart irrigation scheduling for crop production using a crop model and improved deep reinforcement learning, *Agriculture*, 2025, 15(24), 2569.

[14] Y. Chen, M. Lin, Z. Yu, W. Sun, W. Fu, and L. He, Enhancing cotton irrigation with distributional actor-critic reinforcement learning, *Agric. Water Manag.*, 2025, 307, 109194.

[15] X. Ding and W. Du, Optimizing irrigation efficiency using deep reinforcement learning in the field, *ACM Trans. Sens. Netw.*, 2024, 20(4), 99.

[16] J. Zhao, S. Fan, B. Zhang, A. Wang, L. Zhang, and Q. Zhu, Research status and development trends of deep reinforcement learning in the intelligent transformation of agricultural machinery, *Agriculture*, 2025, 15(11), 1223.

[17] 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*, 2023, 23(16), 7128.

[18] A. C. Cob-Parro, Y. Lalangui, and R. Lazcano, Fostering agricultural transformation through AI: An open-source AI architecture exploiting the MLOps paradigm, *Agronomy*, 2024, 14(2), 259.

[19] S. A. Ullah, M. Bibi, S. A. Hassan, H. Abou-Zeid, H. K. Qureshi, H. Jung, A. Mahmood, M. Gidlund, and E. Hossain, From nodes to roads: Surveying DRL applications in MEC-enhanced terrestrial wireless networks, *IEEE Commun. Surv. Tutor.*, 2026, 28, 1169–1208.

[20] R. Finger, J. Sok, E. Ahovi, S. Akter, J. Bremmer, S. Dachbrodt-Saaydeh, C. de Lauwere, C. Kreft, P. Kudsk, F. Lambarraa-Lehnhardt et al., Towards sustainable crop protection in agriculture: A framework for research and policy, *Agric. Syst.*, 2024, 219, 104037.

[21] R. Zhang, H. Zhu, Q. Chang, and M. Mao, A comprehensive review of digital twins technology in agriculture, *Agriculture*, 2025, 15(9), 903.

[22] J. Bellvert, A. Pelechá, M. Pamies-Sans, J. Virgili, M. Torres, and J. Casadesús, Assimilation of Sentinel-2 biophysical variables into a digital twin for the automated irrigation scheduling of a vineyard, *Water*, 2023, 15(14), 2506.

[23] R. G. Alves, R. F. Maia, and F. Lima, Development of a digital twin for smart farming: Irrigation management system for water saving, *J. Clean. Prod.*, 2023, 388, 135920.

[24] 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.*, 2025, 8(2), 38.

Jiamei Liu received the BS degree in automatic engineering from University of Jingdezhen Ceramic, China, in 2023. She is currently pursuing the MS degree in information science and engineering at Zhejiang University, Ningbo, China. Her research interests include deep reinforcement learning and smart agriculture.

Fangle Chang received the PhD degree in agricultural and biological engineering from Pennsylvania State University, USA, in 2016. She is currently an assistant professor in control science and engineering at Ningbo Global Innovation Center, Zhejiang University, Ningbo, China. Her research interests include smart agriculture, complex system control and management, AI in agriculture, and environmental sustainability.

Longhua Ma received the BS degree in industrial electrical automation from Lanzhou Jiaotong University, China, in 1986, and the MS degree in chemical automation engineering and the PhD degree in control science and engineering from Zhejiang University, China, in 1993 and 2002, respectively. From 1986 to 1990, he was an assistant engineer at Qishuyan Locomotive Factory, Changzhou, Jiangsu, China. From 1990 to 2012, he was an associate professor in control science and engineering at Zhejiang University, China. Since 2013, he has been a professor at School of Information Science and Engineering, NingboTech University, China. He has authored 2 books, more than 100 articles, and 10 inventions. His research interests include data mining, system optimization, and intelligent decision-making.

Hongye Su received the BS degree in industrial automation from Nanjing University of Chemical Technology, Jiangsu, China, in 1990, and the MS and PhD degrees in industrial automation from Zhejiang University, Hangzhou, China, in 1993 and 1995, respectively. From 1995 to 1997, he was a lecturer at Department of Chemical Engineering, Zhejiang University, China. From 1998 to 2000, he was an associate professor at Institute of Advanced Process Control, Zhejiang University, China, where he is currently a professor at College of Control Science and Engineering. His research interests include robust control, time-delay systems, and advanced process control theory and applications.