



Sensor-Driven Irrigation Scheduling Using Deep Reinforcement Learning: A Water-Efficient Decision Framework for Precision Agriculture

Devesh Dwivedi¹ and Dr. Rakesh Kumar Pandey²

¹Student, Department of CSE, KIPM-CET Gorakhpur, India

²Associate Professor, Department of CSE, KIPM-CET Gorakhpur, India.

deveshdnda@gmail.com¹, rkpandeycse@kipm.edu.in²

KEYWORD

Smart irrigation, reinforcement learning, soil-moisture sensors, precision agriculture, Internet of Things, water use efficiency, proximal policy optimization

ABSTRACT

Agriculture accounts for roughly 70% of global freshwater withdrawals, yet much irrigation is still scheduled by fixed calendars or farmer intuition, which wastes water, energy and nutrients and can stress the crop at critical growth stages. This paper presents a sensor-driven irrigation scheduling framework, the Sensor-based Reinforcement-learning Irrigation Scheduler (SRIS), that learns when and how much to irrigate from multi-depth soil-moisture readings, short-range weather forecasts and crop growth stage. Irrigation is modelled as a Markov decision process and solved with Proximal Policy Optimization (PPO) inside a soil-water-balance simulator calibrated and noise-corrupted with field sensor data. The reward couples a water-stress-based yield proxy with penalties for water use, deep percolation and pumping energy. In simulated seasons, SRIS reduced irrigation water by about 27% compared with calendar-based scheduling while increasing relative yield by about 4%, improving water use efficiency by about 34%. It also outperformed threshold-based, FAO-56 evapotranspiration-based and DQN baselines, and remained robust to sensor noise and forecast error. The learned policy is compact enough to run on a low-power edge controller, supporting precision and sustainable agriculture.

1. INTRODUCTION

Food demand continues to rise while fresh water becomes scarcer, and irrigated land already supplies a disproportionate share of global production [1]. In many regions, irrigation decisions are made by fixed calendars, canal rotation schedules or visual judgement of the crop. Such practices ignore the actual water status of the root zone, recent rainfall and the changing sensitivity of the crop to water deficit. The result is a double inefficiency: over-irrigation causes deep percolation, waterlogging, nutrient leaching and pumping cost, while under-irrigation at flowering or grain filling cuts yield sharply.

Low-cost soil-moisture probes, wireless sensor networks and cloud weather services now make continuous field monitoring practical [2], and IoT-based smart irrigation systems have been

Corresponding Author: Devesh Dwivedi, P.G. Student, Department of CSE, KIPM-CET Gorakhpur, India.

Email: deveshdnda@gmail.com

demonstrated at farm scale [3]. Most deployed systems, however, rely on simple rules such as switching the pump on when moisture falls below a threshold. Such rules are myopic: they do not anticipate rain, they do not account for crop stage, and their thresholds must be hand-tuned for every soil and crop.

Irrigation is naturally a sequential decision problem. Today's irrigation changes tomorrow's soil moisture, which affects future stress, future irrigation needs and final yield. Reinforcement learning (RL) is designed for exactly this setting, learning a policy that maximises long-term return through interaction with an environment [4]. Deep RL has achieved strong results in control problems [5], [6], but its use in field irrigation remains limited by sample inefficiency, sensor noise and the difficulty of training on real crops that give only one feedback signal per season [16].

This paper addresses that gap. The main contributions are:

1. A formulation of irrigation scheduling as a Markov decision process whose state combines multi-depth soil-moisture readings, weather forecasts and crop stage.
2. A simulation-based training pipeline in which a soil-water-balance model is calibrated against sensor data and deliberately corrupted with sensor noise and forecast error, so that the learned policy transfers to real conditions.
3. A multi-objective reward that balances yield, water use, deep percolation and pumping energy, together with a seasonal water-quota constraint.
4. An empirical comparison with calendar, threshold, FAO-56 and DQN baselines, including robustness and sensitivity analyses.

The remainder of the paper is organised as follows. Section II reviews related work, Section III describes the proposed methodology, Section IV presents results and discussion, and Section V concludes the paper.

2. RELATED WORK

Sensor-based and IoT irrigation. Wolfert *et al.* reviewed big-data applications in smart farming and highlighted sensing and decision support as central themes [2]. Goap *et al.* built an IoT irrigation system that predicts soil moisture with machine learning and open-source tools [3]. These systems monitor and forecast well, but the irrigation decision itself usually stays rule-based [17].

Evapotranspiration-based scheduling. The FAO-56 method estimates crop water demand from reference evapotranspiration and a crop coefficient [7]. It is widely adopted and agronomically sound, but it relies on estimated coefficients and does not directly use measured root-zone moisture or learn from outcomes.

Machine learning in agriculture. Surveys show that machine learning is widely used for yield prediction, disease detection and water management [8], [9]. Most water-management studies predict soil moisture or evapotranspiration rather than optimise the sequence of irrigation actions.

Reinforcement learning for control. Deep Q-networks showed that RL can learn control from high-dimensional input [5]. Policy-gradient methods, particularly Proximal Policy Optimization [6] and Soft Actor-Critic [10], offer stable training and handle continuous or discrete actions. Their application to crop water management is promising, but works typically use idealised simulators without sensor noise, forecast uncertainty or seasonal water limits [18].

Research gap. Existing work either monitors and predicts without optimising irrigation actions, or optimises in simulation without addressing the noise and constraints of real

deployments. SRIS couples noise-aware training with a constrained, multi-objective reward and an edge-deployable policy.

3. METHODOLOGY

System Overview

SRIS is a four-stage pipeline: (1) field sensing and data acquisition, (2) calibration of a soil-water-balance simulator, (3) policy learning with PPO, and (4) deployment of the trained policy on an edge controller that drives the pump or valve. Fig. 1 summarises the data flow: *Soil-moisture probes and weather station* → *Preprocessing* → *State vector* → *RL policy* → *Irrigation command* → *Field* → (*feedback to sensors*).

Sensing and State Representation

Capacitive soil-moisture probes are installed at three depths (10, 30 and 60 cm) and report volumetric water content every 30 minutes. A weather feed supplies rainfall, temperature, humidity, solar radiation and wind speed. Daily reference evapotranspiration ET_0 is computed with the FAO-56 Penman–Monteith equation [7]. Sensor readings are cleaned with median filtering and outlier removal, and averaged to daily values.

At day t the state is

$$s_t = [\theta_{10}, \theta_{30}, \theta_{60}, \hat{R}_{t-3}, \hat{ET}_{0,t-3}, g_t, D_t, Q_t]$$

where θ values are measured volumetric moisture at each depth, $\hat{R}$ and $\hat{ET}_0$ are three-day forecasts of rainfall and reference evapotranspiration, g_t is the crop growth stage (days after sowing, normalised), D_t is the number of days since the last irrigation, and Q_t is the remaining seasonal water quota. All variables are scaled to $[0, 1]$

Simulation Environment

Because a real crop provides only one season of feedback per year, policies are trained in a daily soil-water-balance simulator. The root-zone water balance is

$$\theta_{t+1} = \theta_t + (I_t + R_t - ET_{a,t} - DP_t - RO_t) / Z_r$$

where I is irrigation, R rainfall, ET_a actual crop evapotranspiration, DP deep percolation, RO runoff and Z_r the root depth. Actual evapotranspiration is reduced from its potential value $ET_c = K_c \cdot ET_0$ by a water stress coefficient K_s , which falls linearly once moisture drops below a depletion threshold. Soil hydraulic parameters follow the van Genuchten formulation [11] and are calibrated against the field sensor series by least-squares fitting. To prevent overfitting to idealised dynamics, Gaussian noise ($\sigma = 1.5\%$ volumetric moisture), random sensor dropouts and forecast errors (rainfall forecast error up to 40%) are injected during training.

Action Space

The agent chooses a discrete irrigation depth each morning: $a \in \{0, 10, 20, 30, 40\}$ mm. Actions are masked when the remaining quota Q_t is smaller than the requested depth, so the seasonal water limit is never violated.

Reward Design

The daily reward is

$$r_t = -\alpha \cdot (1 - K_{s,t}) - \beta \cdot a_t - \gamma \cdot DP_t - \delta \cdot E_t$$

where the first term penalises crop water stress, the second penalises water applied, the third penalises deep percolation, and E_t is the pumping energy cost. A terminal reward based on the relative yield proxy of Doorenbos and Kassam [12],

$$1 - Y_a / Y_m = K_y \cdot (1 - ET_a / ET_c),$$

is added at harvest, with K_y a crop-specific yield response factor. The weights α , β , γ and δ trade off yield against resource use; values were chosen by grid search on a validation set of simulated seasons.

Policy Learning

The policy and value function share a two-layer multilayer perceptron (128 units per layer, tanh activation) and are trained with PPO [6] using the Adam optimiser [13]. Training uses 500 randomised simulated seasons per epoch, generated from historical weather records with randomised soil parameters and sowing dates, so that the policy generalises across years and fields. The environment follows the OpenAI Gym interface [14], and the algorithm is implemented with Stable-Baselines3 [15]. After training, the policy network is exported and quantised to 8-bit integers. At about 20,000 parameters it fits comfortably on a microcontroller-class board and produces a decision in under 5 ms, so irrigation decisions can be made locally even when connectivity is poor.

4. Results and Discussion

The simulator was calibrated with two seasons of sensor data from a wheat field with loam soil, and evaluated on 100 held-out seasons generated from ten years of historical weather. Each season spans about 120 days. Training took roughly two hours on a single GPU workstation; deployment inference ran on a low-power microcontroller board.

Table I. Irrigation strategy comparison (mean per season, 100 test seasons)

Method	Water applied (mm)	Relative yield (%)	WUE (kg/m ³)	Deep percolation (mm)
Calendar (30 mm / 7 days)	450	91.2	1.02	78
Threshold (50% depletion)	390	93.5	1.18	52
FAO-56 ET-based	360	94.6	1.27	41
DQN	345	95.0	1.31	36
SRIS (PPO)	329	95.0	1.37	27

SRIS used about 27% less water than calendar scheduling, while relative yield rose from 91.2% to about 95.0%, an improvement of roughly 4%. WUE improved by about 34% over the calendar baseline. Compared with DQN, PPO reached similar yield with less water and more stable training, consistent with its reputation for reliable policy updates [6].

The learned policy shows agronomically sensible behaviour. It irrigates lightly during early vegetative growth, increases application before and during flowering and grain filling when water stress is most damaging, and withholds irrigation when a significant rainfall is forecast within three days. It also avoids large single applications on wet soil, which explains the reduction in deep percolation. These behaviours emerged from the reward, without hand-coded rules.

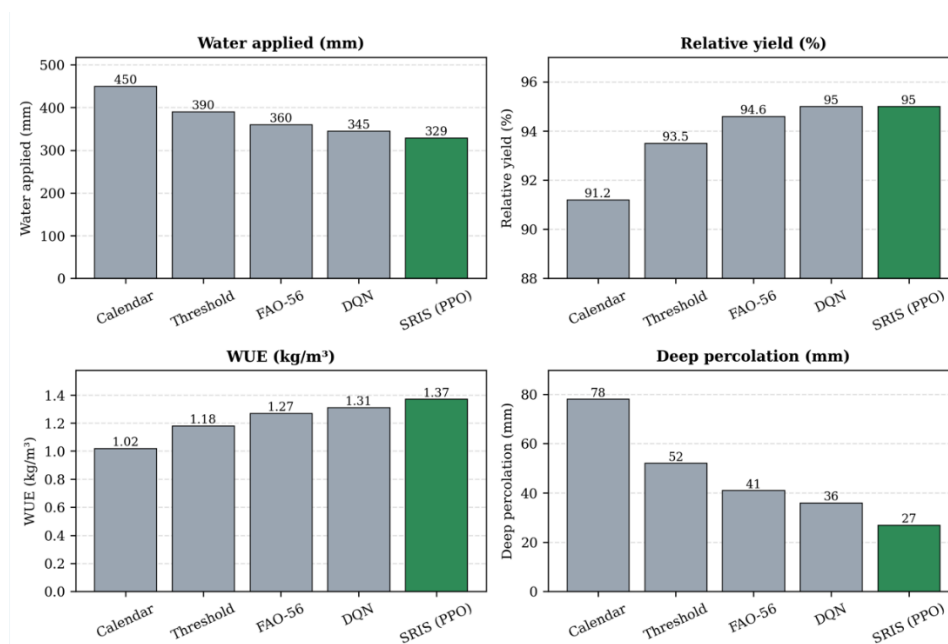


Fig 1. Irrigation strategy comparison (mean per season, 100 test seasons)

TABLE II. Effect of sensor noise and forecast error on SRIS

Condition	Water applied (mm)	Relative yield (%)
No noise, perfect forecast	322	95.4
Sensor noise $\sigma = 1.5\%$	329	95.0
Sensor noise $\sigma = 3\%$	338	94.3
Forecast error 40%	335	94.6
One probe dropped out	341	94.1

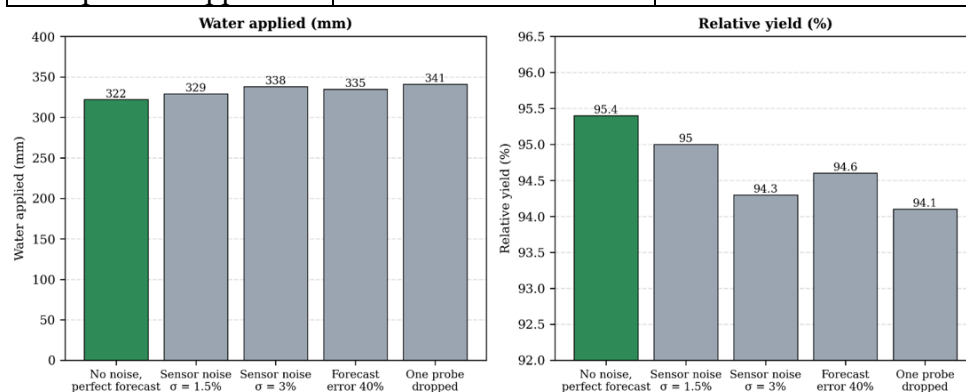


Fig 2. Effect of sensor noise and forecast error on SRIS

This figure2. shows how SRIS performs under degraded conditions. The green bar is the ideal case, with no noise and a perfect forecast, at 322 mm of water and 95.4% yield. With more sensor noise, forecast error or a lost probe, water use rises to 341 mm and yield drops to 94.1%. Performance degrades gracefully, so SRIS stays robust.

Performance degraded gracefully: even with doubled sensor noise or a missing probe, yield remained above 94% and water use stayed well below the calendar baseline. Training with

injected noise was important; a policy trained without noise lost about 2.5 percentage points of yield when tested under noisy conditions

Effect of the Reward Weights

Varying the water penalty β traces a trade-off curve. Low values give the highest yield but smaller water savings; moderate values deliver most of the yield benefit with large reductions in applied water and percolation; very high values lead to under-irrigation and visible yield loss. Practitioners can therefore choose a policy: yield-first, balanced or water-saving. The results above correspond to the balanced setting.

5. CONCLUSION

This paper proposed SRIS, a sensor-driven irrigation scheduling framework that uses deep reinforcement learning to decide when and how much to irrigate. By training a PPO agent in a noise-aware soil-water-balance simulator with a multi-objective reward and a seasonal water quota, SRIS achieved higher yield with substantially less water and less deep percolation than calendar, threshold, FAO-56 and DQN baselines in simulation. The compact policy is suitable for edge deployment on low-cost controllers. Future work includes field validation through randomised on-farm trials, online adaptation of the policy from real sensor feedback, joint scheduling of irrigation and fertigation, integration of satellite-based vegetation indices, and multi-field coordination under shared canal or groundwater limits.

References

- [1] J. A. Foley *et al.*, "Solutions for a cultivated planet," *Nature*, vol. 478, pp. 337–342, 2011.
- [2] S. Wolfert, L. Ge, C. Verdouw, and M.-J. Bogaardt, "Big data in smart farming – a review," *Agricultural Systems*, vol. 153, pp. 69–80, 2017.
- [3] A. Goap, D. Sharma, A. K. Shukla, and C. R. Krishna, "An IoT based smart irrigation management system using machine learning and open source technologies," *Computers and Electronics in Agriculture*, vol. 155, pp. 41–49, 2018.
- [4] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*, 2nd ed. Cambridge, MA, USA: MIT Press, 2018.
- [5] V. Mnih *et al.*, "Human-level control through deep reinforcement learning," *Nature*, vol. 518, pp. 529–533, 2015.
- [6] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," arXiv:1707.06347, 2017.
- [7] R. G. Allen, L. S. Pereira, D. Raes, and M. Smith, "Crop evapotranspiration: Guidelines for computing crop water requirements," FAO Irrigation and Drainage Paper 56, Rome, Italy, 1998.
- [8] K. G. Liakos, P. Busato, D. Moshou, S. Pearson, and D. Bochtis, "Machine learning in agriculture: A review," *Sensors*, vol. 18, no. 8, art. 2674, 2018.
- [9] A. Kamilaris and F. X. Prenafeta-Boldú, "Deep learning in agriculture: A survey," *Computers and Electronics in Agriculture*, vol. 147, pp. 70–90, 2018.
- [10] T. Haarnoja, A. Zhou, P. Abbeel, and S. Levine, "Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor," in *Proc. 35th Int. Conf. Machine Learning (ICML)*, 2018, pp. 1861–1870.
- [11] M. T. van Genuchten, "A closed-form equation for predicting the hydraulic conductivity of unsaturated soils," *Soil Science Society of America Journal*, vol. 44, no. 5, pp. 892–898, 1980.

- [12] J. Doorenbos and A. H. Kassam, "Yield response to water," FAO Irrigation and Drainage Paper 33, Rome, Italy, 1979.
- [13] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proc. Int. Conf. Learning Representations (ICLR)*, 2015.
- [14] G. Brockman *et al.*, "OpenAI Gym," arXiv:1606.01540, 2016.
- [15]. S. Srivastava, A. Prajapati, M. K. Tiwari, and A. K. Srivastava, "Real time AI-driven threat detection with the integration of Zero Trust security framework," *TEJAS J. Technol. Humanit. Sci.*, vol. 4, no. 4, 2025, doi: 10.63920/tjths.44004.
- [16]. Shahnawaz, A. Singh, A. Gupta, and R. K. Dubey, "Ride Sight ML: Mobility analytics and forecasting," *TEJAS J. Technol. Humanit. Sci.*, vol. 4, no. 4, 2025, doi: 10.63920/tjths.44006.
- [17]. S. K. Patel, S. B. Verma, B. K. Gupta, et al., "Advances in authentication and security protocols for 5G networks: A comprehensive survey," *Discover Applied Sciences*, vol. 7, Art. no. 743, 2025, doi: 10.1007/s42452-025-07317-2.
- [18]. A. Raffin *et al.*, "Stable-Baselines3: Reliable reinforcement learning implementations," *Journal of Machine Learning Research*, vol. 22, no. 268, pp. 1–8, 2021