Deep Reinforcement Learning for Irrigation Optimization Based on Crop-Soil Dynamics

Romeo Silvestri, Mattia Antonini, Massimo Vecchio, and Fabio Antonelli
OpenIoT Research Unit, Fondazione Bruno Kessler (FBK), Trento, Italy
Emails: {rsilvestri, m.antonini, mvecchio, fantonelli}@fbk.eu

***Abstract*—Irrigation management is a critical challenge in modern agriculture due to water scarcity and increasing climate variability. This work proposes a deep reinforcement learning (DRL) framework for irrigation scheduling that addresses data scarcity by training the agent within a digital twin of crop–soil dynamics. The environment combines a KNN-based weather generator that extends a 30-year historical record into 1000 synthetic seasons, an XGBoost model for daily soil water tension estimation, and the AquaCrop simulator for crop biomass modeling. A DRL agent is trained using Proximal Policy Optimization to learn a weather-aware irrigation policy without predefined rules. The framework is evaluated on vineyard field data from the Val d'Adige region (Trentino, Italy) over the 2023–2024 growing seasons. Results show a 19% reduction in seasonal water use and more than a twofold increase in the number of days within the optimal soil tension range, while maintaining comparable crop productivity to observed practices.**

***Index Terms*—AquaCrop, irrigation scheduling, simulation-based learning, decision support systems, agricultural digital twin.**

I. Introduction

The growing pressure on global freshwater resources, together with the increasing unpredictability of climate change, has made irrigation management one of the most urgent challenges in modern agriculture. Precision agriculture and the Internet of Things (IoT) have opened the door to data-driven Decision Support Systems (DSS) that aim to improve water use efficiency without sacrificing crop productivity. For irrigation scheduling, these systems have long relied on heuristic approaches or fuzzy-logic controllers [1], [2]. These methods work well when conditions are known in advance, but they have an inherent limitation: the irrigation strategy must be defined *a priori* and encoded as explicit rules or fixed thresholds. When conditions deviate from what was anticipated at design time, the system has no mechanism to adapt.

Digital Twins have recently gained traction as a way to address this gap [3], providing a simulation environment in which strategies can be tested without risking real crops or economic returns. This is where Reinforcement Learning (RL) becomes particularly appealing [4]. Rather than following a predefined strategy, an RL agent learns by interacting with the environment, gradually improving its decisions as it is exposed to varying agronomic and climatic conditions. The application of Deep Reinforcement Learning (DRL) to irrigation scheduling has grown considerably since Yang et al. [5] first demonstrated its feasibility, with subsequent studies addressing a variety of crops and agronomic settings [6]. Kelly et al. [7], for instance, trained a DRL agent within the AquaCrop crop simulator [8], showing that it outperforms conventional heuristics under low rainfall conditions. Saikai et al. [9] reached similar conclusions in a different setting, training an agent on high-dimensional sensor data within the APSIM simulator and outperforming rule-based approaches. Despite these significant advances, training a robust DRL agent requires exposure to a wide and diverse range of conditions, but historical weather records are limited and short-term field observations rarely capture the variability needed to generalize beyond the training period. Data scarcity, in other words, is not a minor inconvenience but a core obstacle to deploying DRL-based irrigation systems in practice.

This paper presents a DRL framework for vineyard irrigation scheduling, developed and validated within an agricultural consortium in the Roverè della Luna municipality (hereafter RdL), in the Val d'Adige region of Trentino, Italy. The framework is built on a digital twin of local crop-soil dynamics, designed to address limited data availability and to bridge the gap between physics-based crop models and real-time field sensor measurements. To this end, four interconnected components are integrated: a K-Nearest Neighbors (KNN) bootstrap resampling procedure [10] for synthetic weather generation, an XGBoost model for daily soil water tension estimation, the FAO AquaCrop model [8] for crop growth simulation, and a Proximal Policy Optimization (PPO) agent with Generalized Advantage Estimation (GAE) [11] for irrigation policy learning.

The remainder of this paper is organized as follows. Sec. II describes the proposed methodology and details the components of the digital twin architecture. Sec. III presents the simulation pipeline and the agent training procedure. Sec. IV reports experimental results. Finally, Sec. V draws conclusions and outlines directions for future work.

II. Proposed Methodology

The proposed methodology consists of multiple components, as illustrated in Fig. 1. First, a KNN bootstrap generates synthetic multi-year weather sequences from historical climate records, extending the range of scenarios encountered during training. Soil water tension is estimated daily using a machine learning (ML) model, selected among several candidates via expanding window cross-validation. Crop growth under varying irrigation strategies is simulated using AquaCrop. A DRL agent is then trained within this environment using PPO, learning a policy that balances crop productivity and water consumption. The following subsections describe each component in detail.

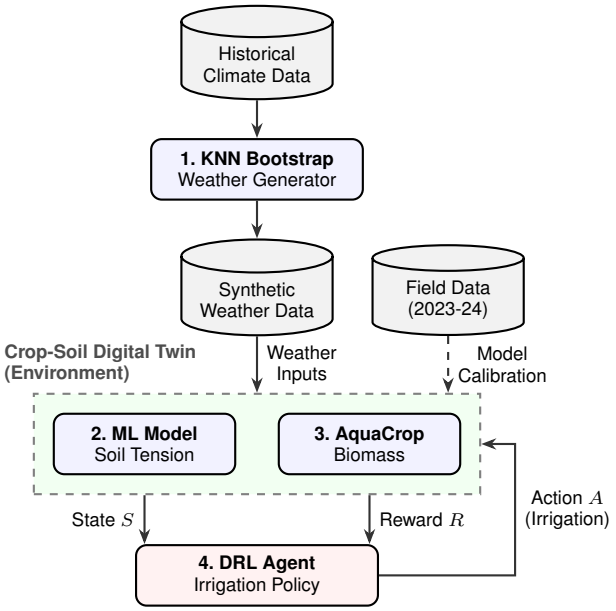

Fig. 1. Overall architecture of the proposed framework. Rectangles represent processing components; cylinders represent data sources.

A. Climate Data Acquisition and Scenario Generation

Let $\mathcal{D} = \{(\mathbf{x}_i, d_i)\}_{i=1}^{N}$ denote the historical dataset, where N is the number of observations, $\mathbf{x}_i \in \mathbb{R}^p$ is the meteorological feature vector associated with time step i, and $d_i \in \{1, \ldots, 366\}$ denotes the corresponding day of the year. Each vector $\mathbf{x}_i$ comprises both current observations and their lagged values over the preceding 3 days. In the present study, $\mathcal{D}$ was assembled using data obtained from OpenMeteo (https://open-meteo.com), an open-access API providing historical and forecast weather information for user-specified geographic coordinates.

Meteorological records were extracted for the RdL site, located at $46.2501°$ N and $11.1723°$ E, over the 30-year period from January 1, 1990 to December 31, 2019. This period was selected for two reasons. First, it provides a sufficiently long observational record to support a robust characterization of local weather variability for simulation purposes. Second, it corresponds to a standard climatological reference period, thereby ensuring consistency with the baseline climatic conditions representative of recent decades. All variables were aggregated at daily resolution and included FAO reference evapotranspiration (ET_0), cumulative precipitation (P^{tot}), mean relative humidity at 2 m above ground (RH), minimum and maximum air temperature at 2 m ($T^{\min}$ and $T^{\max}$), accumulated shortwave radiation (SR), and mean wind speed at 10 m (W). These variables were retained because they represent the dominant meteorological controls on soil water tension and, more generally, on soil moisture dynamics.

For synthetic weather generation, we adopted the KNN bootstrap, a well-established nonparametric approach for multivariate daily weather generation [12]. This method was selected because it can generate synthetic sequences without requiring parametric assumptions on the joint distribution of the variables, while naturally preserving cross-correlations and the temporal dependence structure observed in the historical record.

The KNN bootstrap procedure is formalized in Algorithm 1, which describes how, given a window size $w \in \mathbb{N}$ and a number of neighbors $k \in \mathbb{N}$, a synthetic weather dataset of length T is generated by iteratively sampling successor states. In this study, the parameters were set to $w = 15$ days and $k = 5$ neighbors, providing a balance between variability and fidelity to the observed record. The seasonal window $\mathcal{W}(d_t, w)$ restricts the neighbor search to historically comparable days, so that the simulated sequence respects the structure of the observed record. Year boundaries are handled with circular continuity to ensure seamless transitions between December and January. The use of successor records rather than the sampled ones themselves ensures temporal continuity and realistic day-to-day transitions in the generated weather data. At each iteration, the lag values of all variables are updated to maintain internal consistency throughout the generated trajectory. Leap years in the synthetic series are determined by mapping each simulated year to the Gregorian calendar cycle. By repeatedly applying the KNN bootstrap, the historical record was expanded into a synthetic dataset of 1000 simulated years.

Algorithm 1 KNN Bootstrap Resampling for Weather Data

Require: Historical dataset $\mathcal{D} = \{(\mathbf{x}_i, d_i)\}_{i=1}^{N}$; window size w; neighbors k; simulation horizon T

Ensure: Simulated weather dataset $\{\hat{\mathbf{x}}_t\}_{t=1}^{T}$

Initialize $\hat{\mathbf{x}}_1$ by sampling uniformly from $\mathcal{W}(1, w)$

for $t = 1, \ldots, T-1$ **do**

Step 1: Seasonal window selection

$$\mathcal{W}(d_t, w) \leftarrow \big\{ i \in \{1, \ldots, N-1\} \mid \min\big(|d_i - d_t|, 366 - |d_i - d_t|\big) \le w \big\}$$

Step 2: Nearest neighbor search

Compute Euclidean distance in standardized space:

$$\delta_i \leftarrow \|\hat{\mathbf{z}}_t - \mathbf{z}_i\|_2 \quad \forall\, i \in \mathcal{W}(d_t, w)$$

with $\mathbf{z} = (\mathbf{x} - \boldsymbol{\mu})/\boldsymbol{\sigma}$

Retain the k candidates with the smallest distance:

$$\mathcal{N}_k(\hat{\mathbf{z}}_t) \leftarrow \underset{i \in \mathcal{W}(d_t, w)}{\arg\min_k}\ \delta_i$$

Step 3: Rank-based weighted sampling

Sort the k neighbors and assign to the j-th a weight:

$$\tilde{p}_j \leftarrow \frac{1/j}{\sum_{l=1}^{k} 1/l}, \quad j = 1, \ldots, k$$

Sample one neighbor i^* from the k nearest:

$$i^* \sim \mathrm{Categorical}(\tilde{p}_1, \ldots, \tilde{p}_k)$$

Step 4: Successor extraction

$$\hat{\mathbf{x}}_{t+1} \leftarrow \mathbf{x}_{i^*+1}$$

Step 5: Lag update

Replace lags in $\hat{\mathbf{x}}_{t+1}$ with previously simulated states $\{\hat{\mathbf{x}}_t, \hat{\mathbf{x}}_{t-1}, \hat{\mathbf{x}}_{t-2}\}$

end for

B. Soil Tension Estimation via Machine Learning

Soil water tension at 30 cm depth was estimated using an ML model trained on data collected during the growing seasons of 2023 and 2024 in RdL. The feature set includes meteorological variables retrieved from OpenMeteo, their lagged values over the preceding 3 days, daily irrigation records in mm, and soil tension measurements in mbar collected from 8 tensiometers and aggregated to a daily mean. Including lagged soil tension values as predictors endows the model with an autoregressive structure, enabling it to capture the temporal dynamics of soil moisture depletion and recharge.

The dataset spans from April 21, 2023 to October 24, 2024, for a total of 553 daily observations divided into three partitions. The training set covers the period from April 21, 2023 to April 19, 2024 (365 samples), the validation set from April 20 to July 31, 2024 (103 samples), and the test set from August 1 to October 24, 2024 (85 samples). Each partition includes periods with and without irrigation events to cover the full range of agronomic conditions.

Model selection was performed via expanding-window cross-validation (hereafter, rolling validation): starting with an initial training window, the model is retrained daily on all data available up to the current date and evaluated the following day. This procedure provides a realistic estimate of out-of-sample performance under the temporal structure of the data. Hyperparameter optimization was carried out using Optuna [13] over 100 trials, adopting a three-objective formulation that jointly minimizes RMSE and MAE while maximizing R^2. The search was performed using the NSGA-II evolutionary algorithm [14] to identify the Pareto front of non-dominated configurations. The best configuration was selected as the point on the Pareto front minimizing the Euclidean distance from the ideal optimum in the normalized space. Specifically, each objective was rescaled to $[0, 1]$ via min-max normalization, with R^2 additionally inverted so that lower values correspond to better performance; the ideal point thus corresponds to the origin of this normalized space. As a supplementary analysis, a seasonal simulation was conducted to assess each model's ability to sustain accurate predictions over long time horizons without saturation, divergence, or error accumulation. Given the limited dataset size, this evaluation was performed across all three data partitions, intentionally introducing a degree of data leakage. This analysis is therefore used as a secondary selection criterion alongside the rolling validation. Several models were evaluated, including a linear autoregressive model with exogenous variables (ARX), Support Vector Regression (SVR), Random Forest, and XGBoost. Hyperparameter optimization was applied to all models except ARX, whose structure is fixed. The search ranges adopted are reported in Table I.

TABLE I
HYPERPARAMETER SEARCH RANGES FOR EACH CANDIDATE MODEL.

Model	Hyperparameter	Type	Range / Values
SVR	`kernel`	Categorical	{rbf, poly}
	`C`	Log-uniform	[0.01, 100]
	`epsilon`	Log-uniform	[0.001, 10]
	`gamma`	Categorical	{scale, auto}
	`degree`	Integer	[2, 4] (poly only)
Random Forest	`n_estimators`	Integer	[100, 500]
	`max_depth`	Integer	[3, 15]
	`min_samples_split`	Integer	[2, 20]
	`min_samples_leaf`	Integer	[1, 10]
	`max_features`	Uniform	[0.3, 1.0]
	`bootstrap`	Categorical	{True, False}
XGBoost	`n_estimators`	Integer	[100, 500]
	`max_depth`	Integer	[3, 7]
	`min_child_weight`	Integer	[3, 10]
	`learning_rate`	Log-uniform	[0.01, 0.3]
	`subsample`	Uniform	[0.5, 1.0]
	`colsample_bytree`	Uniform	[0.5, 1.0]
	`gamma`	Log-uniform	[0.001, 1.0]
	`reg_alpha`	Log-uniform	[0.001, 1.0]
	`reg_lambda`	Log-uniform	[0.001, 1.0]

C. The AquaCrop Simulation Environment

AquaCrop is a crop water productivity model developed by the Food and Agriculture Organization of the United Nations (FAO) in 2009 for the simulation of crop yield response and water use under varying management and environmental conditions [8]. In the present work, AquaCrop was configured to represent the specific agronomic and pedological conditions of RdL. Soil parameters were derived from local field measurements conducted in the study area, while crop parameters were calibrated for the Lagrein grapevine variety by combining domain knowledge provided by local agronomists with values reported in the literature for comparable cultivars [15]. The key calibrated parameters include water productivity ($WP = 35$ g m^{-2}), maximum canopy cover ($CC_x = 0.80$), basal crop coefficient ($K_{cb} = 0.60$), and maximum rooting depth ($Z_{\max} = 1.2$ m). Phenological development was defined using calendar days, with emergence on April 20, flowering spanning late May to early June, and physiological maturity on September 20.

The model operates at a daily time step and receives as input ET_0, $T^{\min}$, $T^{\max}$, and P^{tot}, consistent with the variables retrieved from OpenMeteo. Given a sequence of daily irrigation decisions $\{I_t\}_{t=1}^{N}$, AquaCrop yields daily estimates of aboveground biomass B_t and potential biomass $B_t^{\max}$, where the latter represents the theoretical maximum accumulation under non-limiting water conditions and serves as a normalization reference in the reward function of Eq. (2).

D. Reinforcement Learning Formulation

The irrigation scheduling problem is formulated as a Markov Decision Process (MDP) $\langle \mathcal{S}, \mathcal{A}, \mathcal{T}, \mathcal{R} \rangle$, where $\mathcal{S}$ is the state space, $\mathcal{A}$ is the action space, $\mathcal{T}$ is the stochastic transition dynamics defined by the digital twin as a distribution over next states, and $\mathcal{R}$ is the reward function. The goal of the agent is to learn a policy $\pi_\theta : \mathcal{S} \to \mathcal{A}$ that maximizes the expected cumulative reward over a growing season.

State space. At each day t, the state vector is defined as:

$$\mathbf{S}_t = \left(\bar{\phi}_t,\, \hat{\phi}_{t+1}^{(0)},\, \hat{P}_{t,3}^{\mathrm{tot}},\, \hat{T}_{t,3}^{\max} \right) \tag{1}$$

where $\bar{\phi}_t$ is the average soil water tension previously estimated by the ML model at day t; $\hat{\phi}_{t+1}^{(0)}$ is the predicted

average soil tension for the following day under the assumption of no irrigation; $\hat{P}_{t,3}^{\text{tot}}$ and $\hat{T}_{t,3}^{\max}$ are the forecasted cumulative precipitation and maximum temperature over the next 3 days, respectively. This state representation provides the agent with both the current soil water status and short-term meteorological information, enabling anticipatory irrigation decisions.

Action space. The action $a_t \in [0, I_{\max}]$ represents the amount of water (in mm) decided on day t and to be delivered on day $t+1$, where $I_{\max}$ is the maximum irrigation volume deliverable in a single day. The action space is continuous, and the policy π_θ outputs a distribution over this interval from which the irrigation quantity is sampled during training and taken deterministically at inference time.

Reward function. The reward R is designed to jointly incentivize crop productivity, water use efficiency, and maintenance of optimal soil water conditions:

$$R = w_1 \cdot \frac{B}{B_{\max}} - w_2 \cdot \frac{I}{I_{\max}} - w_3 \cdot g(\phi) \tag{2}$$

where B and $B_{\max}$ are the daily biomass and potential biomass estimated by AquaCrop, I is the irrigation action suggested by the agent, $I_{\max}$ is the maximum irrigation volume deliverable in a single day, and $g(\phi)$ is a penalty term penalizing deviations of the soil tension from the agronomically optimal range. The penalty function is defined as:

$$g(\phi) = \begin{cases} 0 & \text{if } \phi_{\min} \le \phi \le \phi_{\max} \\ \dfrac{\phi_{\min} - \phi}{\phi_{\text{scale}}} & \text{if } \phi < \phi_{\min} \\ \dfrac{\phi - \phi_{\max}}{\phi_{\text{scale}}} & \text{if } \phi > \phi_{\max} \end{cases} \tag{3}$$

where $\phi_{\min}$ and $\phi_{\max}$ define the optimal tensiometric range, and ϕ_{scale} is a normalization factor. For the grapevine crop considered in this study, these are set to $\phi_{\min} = 200$ mbar (wet threshold), $\phi_{\max} = 400$ mbar (dry threshold), and $\phi_{\text{scale}} = 200$ mbar. The weights w_1, w_2, w_3 allow tuning the trade-off between the components of the reward; in this work, we arbitrarily choose $w_1 = w_3 = 1$ and $w_2 = 0.5$, prioritizing crop productivity and avoidance of soil stress over water saving. A systematic tuning of these weights is beyond the scope of this paper and is left as future work.

Optimization algorithm. The policy π_θ is optimized using PPO [11], an actor-critic algorithm that constrains policy updates via a clipped surrogate objective. Policy and value networks are parameterized as multi-layer perceptrons with two hidden layers of 64 neurons. The full training procedure is detailed in Algorithm 2.

III. Simulation Pipeline

Algorithm 2 details the training loop. At each step, the ML model provides a counterfactual soil tension estimate assuming no irrigation, which the agent uses to construct the state vector and output an irrigation recommendation. If irrigation is prescribed, the soil tension estimate is updated accordingly. AquaCrop is subsequently queried to estimate the expected biomass. The reward is then computed, and the resulting transition is stored in a buffer and used at the end of each season to update the policy via GAE and PPO.

Algorithm 2 DRL loop for Irrigation Scheduling

Require: Simulated weather $\{\hat{\mathbf{x}}_t\}_{t=1}^{T}$ ($T = Y \times D$); ML model $f(\cdot)$; AquaCrop simulator; reward weights (w_1, w_2, w_3)
Ensure: Trained irrigation policy π_{θ^*}

for $y = 1, \dots, Y$ **do**
  Initialize $\bar{\phi}_1$ and $\mathbf{S}_1$ from initial field conditions
  for $t = 1, \dots, D-1$ **do**
    Step 1: Soil tension simulation without irrigation
      $\hat{\phi}_{t+1}^{(0)} \leftarrow f(\hat{\mathbf{x}}_t, \{\bar{\phi}_{t-j}\}_{j=0}^{2}, I_{t+1} = 0)$
    Step 2: Irrigation suggestion
      Compute state vector:
      $\mathbf{S}_t \leftarrow (\bar{\phi}_t, \hat{\phi}_{t+1}^{(0)}, \hat{P}_{t,3}^{\text{tot}}, \hat{T}_{t,3}^{\max})$
      Sample irrigation action from policy:
      $I_{t+1} \sim \pi_\theta(\cdot \mid \mathbf{S}_t)$
    Step 3: Soil tension and biomass estimation
    if $I_{t+1} > 0$ **then**
      $\hat{\phi}_{t+1} \leftarrow f(\hat{\mathbf{x}}_t, \{\bar{\phi}_{t-j}\}_{j=0}^{2}, I_{t+1})$
    else
      $\hat{\phi}_{t+1} \leftarrow \hat{\phi}_{t+1}^{(0)}$
    end if
      $\bar{\phi}_{t+1} \leftarrow \hat{\phi}_{t+1}$
      Predict biomass $\hat{B}_{t+1}$ and $\hat{B}_{t+1}^{\max}$ with AquaCrop
    Step 4: Reward estimation and storing
      $R_t \leftarrow w_1 \cdot \frac{\hat{B}_{t+1}}{B_{t+1}^{\max}} - w_2 \cdot \frac{I_{t+1}}{I_{\max}} - w_3 \cdot g(\hat{\phi}_{t+1})$
      $\mathbf{S}_{t+1} \leftarrow (\bar{\phi}_{t+1}, \hat{\phi}_{t+2}^{(0)}, \hat{P}_{t+1,3}^{\text{tot}}, \hat{T}_{t+1,3}^{\max})$
      Store transition $(\mathbf{S}_t, I_{t+1}, R_t, \mathbf{S}_{t+1})$ in buffer
  end for
  Step 5: Policy update
    Compute advantage estimates via GAE
    Optimize π_θ via PPO
    Clear the buffer
end for

IV. Analysis and Results

The KNN bootstrap reproduces the statistical properties of the historical climate record. To formally assess distributional similarity, we applied the two-sample Kolmogorov-Smirnov (KS) test, which measures the maximum difference between two empirical cumulative distribution functions. Since the synthetic dataset is substantially longer than the historical record, direct comparison would yield artificially significant results. We therefore repeatedly sampled 100 sequences of 30-year length from the synthetic data and report the median KS statistic and p-value across subsamples. Table II shows that mean and standard deviation differ by less than 2% across all

variables. Formal KS tests confirm no statistically significant difference ($p > 0.05$) for ET_0, precipitation, shortwave radiation, and temperature, while relative humidity and wind speed exhibit statistically significant differences ($p < 0.001$), likely attributable to minor distributional shifts introduced by the resampling process rather than systematic bias, as the mean deviations remain below 1%. Additional diagnostics, including pairwise correlation matrices, seasonal cycle of mean values, and dry spell length, confirmed results consistent with those reported in the literature for KNN-based generators [12].

TABLE II
SUMMARY STATISTICS OF HISTORICAL AND SYNTHETIC WEATHER DATA.

	Mean ± Std. Dev.		**KS Test**	
Variable	Hist.	Synt.	Stat.	p-value
ET_0 (mm)	2.31 ± 1.45	2.29 ± 1.44	0.012	0.430
Precipitation (mm)	2.49 ± 6.14	2.41 ± 5.99	0.009	0.737
Rel. Humidity (%)	69.38 ± 12.15	70.03 ± 11.70	0.031	<0.001
Min Temperature (°C)	8.24 ± 7.79	8.50 ± 7.69	0.018	0.065
Max Temperature (°C)	16.91 ± 8.03	17.14 ± 7.95	0.017	0.090
SW Radiation ($W\,m^{-2}$)	13.27 ± 6.87	13.13 ± 6.80	0.013	0.318
Wind Speed ($m\,s^{-1}$)	4.38 ± 1.35	4.30 ± 1.30	0.030	<0.001

Table III reports the performance of all candidate models under both evaluation procedures. As a reference, a naive persistence baseline is included, which predicts the next-day soil tension as equal to the current observed value. In the rolling validation, ARX and XGBoost achieve comparable results, with RMSE of 28.30 and 28.76 respectively, both outperforming the persistence baseline (RMSE of 34.77) as well as SVR and Random Forest. However, the seasonal simulation reveals a substantial divergence: ARX degrades significantly over long time horizons, reaching an RMSE of 121.01 and an R^2 of 0.364. Simulated trajectories reveal that the ARX model tends to over-smooth the predicted series, failing to reproduce the sharp tension peaks that follow dry spells and the rapid drops triggered by irrigation or precipitation events. XGBoost, by contrast, achieves an RMSE of 90.30 and an R^2 of 0.646, confirming its ability to capture the nonlinear dynamics of soil water tension and generalize across extended simulation horizons. Based on this combined evaluation, XGBoost was selected as the final soil tension model and integrated as the simulation component within the DRL training loop.

TABLE III
PERFORMANCE METRICS OF ML MODELS.

	Rolling validation			**Seasonal simulation**		
Model	RMSE	MAE	R^2	RMSE	MAE	R^2
Naive baseline	34.77	22.10	0.916	–	–	–
ARX	28.30	17.43	0.944	121.01	80.93	0.364
SVR	52.40	28.44	0.808	102.84	38.21	0.541
Random forest	34.08	19.60	0.919	101.23	64.66	0.555
XGBoost	28.76	16.38	0.942	90.30	45.64	0.646

The DRL agent was then trained on the 1000-year synthetic weather dataset following a two-stage learning strategy. In the first stage, the agent was trained using only the water saving and soil tension terms of the reward, setting $w_1 = 0$. This allowed the agent to first learn the relationship between irrigation decisions and soil water dynamics, without the additional complexity introduced by the biomass term. In the second stage, the full reward was activated by restoring $w_1 = 1$, allowing the agent to refine its policy by incorporating crop productivity. This approach was motivated by the observation that activating all three reward terms from the start caused convergence failure, with increasing policy variance and low explained variance throughout training, suggesting entrapment in a local optimum. The two-stage approach resolved this issue, enabling stable and consistent convergence across training runs. Training curves are omitted for brevity and will be reported in an extended version of this work.

In the comparison with the observed field irrigation schedule, the agent adopts a qualitatively distinct irrigation strategy. As visible in Fig. 2, the real schedule follows a threshold-based logic, applying a small number of large irrigation events designed to drive soil tension from the dry threshold back below the wet threshold, resulting in wide oscillations of the tensiometer. The DRL agent, by contrast, distributes irrigation across multiple days at low intensity, with most events in the range of 0.5–1 mm, maintaining soil tension within a narrower and more stable band. Furthermore, by exploiting the 3-day precipitation forecast included in the state representation, the agent reduces irrigation in the days preceding rainfall events, enabling weather-aware regulation that is only partially reflected in observed field practice.

These behavioral differences are reflected in Table IV. In 2023 the agent recommended more irrigation than observed in the field ($+50.5\%$), while in 2024 it applied considerably less (-58.6%), resulting in a seasonal mean water saving of 19.2%. In terms of expected biomass simulated by AquaCrop, the real schedule yields higher values on average (19.92 vs. 19.76 t/ha), though the difference is minimal and both scenarios approach the potential maximum. The most substantial improvement concerns soil tension regulation: the number of days within the optimal tensiometric range is more than doubled on average (67.5 vs. 29.5 per season), indicating that the learned policy maintains crop water status within favorable conditions more consistently than the observed field practice.

V. CONCLUSIONS

This paper presented a DRL framework for vineyard irrigation scheduling built upon a digital twin of the crop-soil system. The proposed approach addresses the challenge of limited data availability by grounding agent training in synthetic growing seasons generated from historical climate records.Simulation results show that the learned policy reduces seasonal water consumption by approximately 19% while more than doubling the number of days within the optimal soil tension range, with only a marginal reduction in AquaCrop-estimated biomass compared to current field practice. Future work will focus on field validation under operational conditions, comparison against established irrigation decision-support baselines, systematic

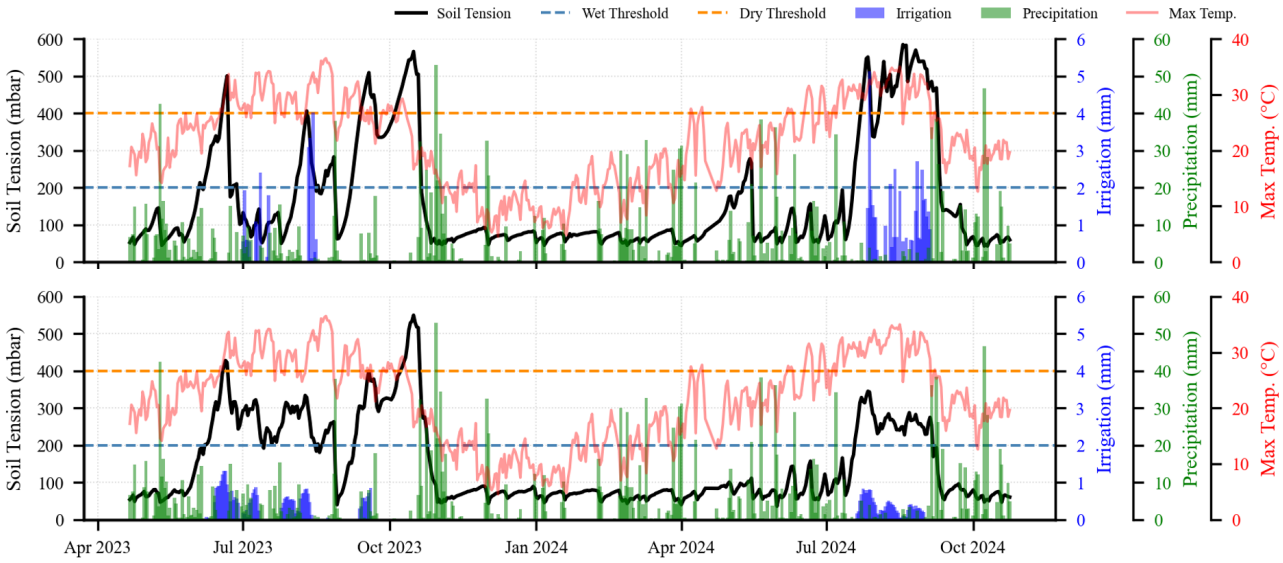

Fig. 2. Comparison of actual and DRL-simulated irrigation schedules for RdL during 2023 and 2024. The upper graph shows the real tensiometer readings and irrigation events, while the lower graph displays the simulated soil tension derived from the RL agent recommendations.

TABLE IV
PERFORMANCE METRICS COMPARISON: REAL VS. DRL AGENT.

	2023			2024			Seasonal Mean		
Metric	Real	DRL Agent	Δ(%)	Real	DRL Agent	Δ(%)	Real	DRL Agent	Δ(%)
Total irrigation (mm)	22.04	33.18	+50.5	38.96	16.13	−58.6	30.50	24.66	−19.2
Mean soil tension (mbar)	178.31	203.58	+14.2	242.41	148.59	−38.7	210.36	176.08	−16.3
Days in optimal soil tension	41	86	+109.8	18	49	+172.2	29.5	67.5	+128.8
Cumulative biomass (t/ha)	19.92	19.92	0.0	19.92	19.61	−1.6	19.92	19.76	−0.8

tuning of the reward weights, and extension to additional crop varieties and field sites.

ACKNOWLEDGMENT

This work was partially supported by the IT4LIA AI FACTORY project, funded by the Italian Ministry of Universities and Research, the National Cybersecurity Agency, and the European Commission through the European High Performance Computing Joint Undertaking (GA #101234224).

REFERENCES

[1] R. Silvestri, M. Vecchio, and F. Antonelli, "A fuzzy decision support system to optimize irrigation practices in trentino region," in *2025 11th International Conference on Control, Decision and Information Technologies (CoDIT)*, vol. 1. IEEE, 2025, pp. 1–6.

[2] R. Silvestri, M. Vecchio, M. Pincheira, and F. Antonelli, "Smart irrigation with fuzzy decision support systems in trentino vineyards," *Sensors*, vol. 25, no. 23, p. 7188, 2025.

[3] G. Goldenits, K. Mallinger, S. Raubitzek, and T. Neubauer, "Current applications and potential future directions of reinforcement learning-based digital twins in agriculture," *Smart Agricultural Technology*, vol. 8, p. 100512, 2024.

[4] H. Overweg, H. N. Berghuijs, and I. N. Athanasiadis, "Cropgym: a reinforcement learning environment for crop management," *arXiv preprint arXiv:2104.04326*, 2021.

[5] Y. Yang, J. Hu, D. Porter, T. Marek, K. Heflin, and H. Kong, "Deep reinforcement learning-based irrigation scheduling," *Transactions of the ASABE*, vol. 63, no. 3, pp. 549–556, 2020.

[6] M. Chen, Y. Cui, X. Wang, H. Xie, F. Liu, T. Luo, S. Zheng, and Y. Luo, "A reinforcement learning approach to irrigation decision-making for rice using weather forecasts," *Agricultural Water Management*, vol. 250, p. 106838, 2021.

[7] T. D. Kelly, T. Foster, and D. M. Schultz, "Assessing the value of deep reinforcement learning for irrigation scheduling," *Smart Agricultural Technology*, vol. 7, p. 100403, 2024.

[8] P. Steduto, T. C. Hsiao, D. Raes, and E. Fereres, "Aquacrop—the fao crop model to simulate yield response to water: I. concepts and underlying principles," *Agronomy journal*, vol. 101, no. 3, pp. 426–437, 2009.

[9] Y. Saikai, A. Peake, and K. Chenu, "Deep reinforcement learning for irrigation scheduling using high-dimensional sensor feedback," *PLoS Water*, vol. 2, no. 9, p. e0000169, 2023.

[10] U. Lall and A. Sharma, "A nearest neighbor bootstrap for resampling hydrologic time series," *Water resources research*, vol. 32, no. 3, pp. 679–693, 1996.

[11] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," *arXiv preprint arXiv:1707.06347*, 2017.

[12] B. Rajagopalan and U. Lall, "A k-nearest-neighbor simulator for daily precipitation and other weather variables," *Water resources research*, vol. 35, no. 10, pp. 3089–3101, 1999.

[13] T. Akiba, S. Sano, T. Yanase, T. Ohta, and M. Koyama, "Optuna: A next-generation hyperparameter optimization framework," in *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*, 2019, pp. 2623–2631.

[14] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: Nsga-ii," *IEEE transactions on evolutionary computation*, vol. 6, no. 2, pp. 182–197, 2002.

[15] S. Er-Raki, E. Bouras, J. Rodriguez, C. Watts, C. Lizarraga-Celaya, and A. Chehbouni, "Parameterization of the aquacrop model for simulating table grapes growth and water productivity in an arid region of mexico," *Agricultural Water Management*, vol. 245, p. 106585, 2021.