Learning Extended Forecasts of Soil Water Content via Physically-Inspired Autoregressive Models

Ozmen Erkin Kokten[1], Raviv Raich[1], James Holmes[2], and Alan Fern[1]
[1]Oregon State University, Corvallis, OR 97331-5501 [2]Ciel du Cheval Vineyard, West Richland, WA.
Email: {kokteno, raviv.raich, alan.fern}@oregonstate.edu, jimgrape@msn.com

***Abstract*—Vine stress resulting from soil water content (SWC) restrictions allows growers to improve grape and subsequent wine quality. In this work, we consider learning models that can forecast SWC to assist growers' irrigation decisions. In particular, we investigate training auto-regressive recurrent neural networks to make multi-day hourly forecasts of SWC based on historical data from soil-moisture sensors, irrigation schedules, and evapotranspiration estimates. Our work addresses two practical challenges in training such models. First, trained autoregressive models are prone to error propagation, which quickly degrades longer-term forecasts. Second, it is difficult to learn the underlying causal relationship between irrigation and soil moisture due to the training data having limited coverage of the primary control input, irrigation. We propose a training strategy that combines one-step teacher forcing loss with a loss over multi-step autoregressive predictions and novel regularization terms to ensure SWC forecasts align with scientific models, effectively addressing the key challenges. We present results from five irrigation blocks with two cultivars, using datasets ranging from 2947 to 4784 hourly measurements of SWC, irrigation, and weather. Our methodology achieves precise SWC predictions and generates realistic forecasts for untrained irrigation scenarios.**

***Index Terms*—time-series, teacher-forcing, autoregressive training, non-linear state-space models, smart agriculture**

I. Introduction

Controlled water stress for grapevines involves limiting the supply of water during grape development in order to promote increased sugar production, which can translate to higher-quality grapes and wine. A common challenge for growers is to balance between keeping the soil moisture levels sufficiently low to induce water stress, but high enough to avoid plant wilting. In this paper, we study the use of machine learning for soil water content forecasting to assist growers in making decisions about irrigation control to manage stress.

As illustrated in Fig. 1, our application scenario involves grapevines with nearby soil water content (SWC) sensors at different depths. We focus on multi-day forecasting of hourly SWC values based on the SWC history, weather forecast, and planned irrigation schedules. This capability can be used by growers to forecast alternative irrigation schedules to achieve desired levels of water stress. Achieving accurate forecasts, however, is challenging due to the complex interplay between the unknown soil and root structure in the vicinity of the plant.

This work was partially supported by USDA NIFA award No. 2021-67021-35344 (AgAID AI Institute).

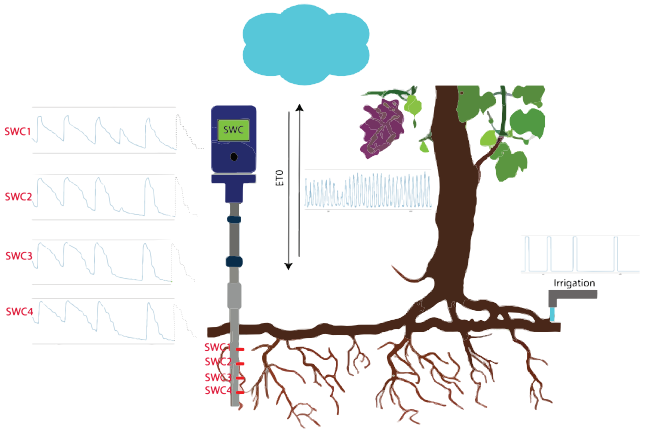

Fig. 1. Problem setup at a high level. Soil water content (SWC) measurements at multiple depth levels in the vicinity of the grapevine's root system are collected. Additionally, reference evapotranspiration (ET_0) and on/off irrigation information are available. Given a future irrigation plan, the goal is to forecast SWC values. OpenAI's ChatGPT was used to generate this figure.

One approach to forecasting is physics-based modeling. To capture soil behavior, nonlinear models [1] and dependence on environmental factors [2] are considered. Finite-element models of water flow, such as HYDRUS [3] can offer accurate SWC forecasts assuming voxel-level information about soil composition and plant-root structure. Since it is impractical to obtain such voxel-level information outside of a laboratory, we instead focus on a data-driven approach. In particular, for each SWC sensor array, we use its historical data to learn a predictive model of the nonlinear dynamics that capture the plant-soil response to irrigation and weather effects. This allows for customized predictions across different soil and root structures without explicitly knowing those structures. Data-driven state-space models such as recurrent neural nets (RNN) have been considered in the realm of time series forecasting [4]. LSTM and Bi-LSTM models have been used to perform short-term forecast soil moisture levels utilizing soil and weather data, with the aim of enhancing smart irrigation systems to prevent drought stress on crops [5].

Our main contribution is a new training procedure for learning non-linear state-space models for long-term SWC forecasting using autoregressive methods, where model predictions are recursively inputted for extended forecasts. This approach introduces error-propagation issues where small in-

dividual prediction errors compound to yield large errors in long-term forecasts. To address this challenge, we introduce a training loss function that combines a *teacher-forcing* term and an autoregressive term to make accurate one-step predictions as well as minimize the impact of error propagation. We show that the inclusion of both of these terms is important for obtaining fast learning and stability of long-term predictions. A second challenge is to learn physically plausible models, especially given the limited range of irrigation schedules in our training data. To that end, we introduced a novel regularization term that penalizes forecasts that violate basic water conservation principles. By training on simulated irrigation patterns outside of the training data, the regularization results in models that reliably produce physically plausible forecasts. To assess our contribution, we conducted experiments on five different irrigation blocks involving two grape cultivars of the Ciel du Cheval vineyard located in eastern Washington. Overall, our results on this data indicate that our proposed model is able to improve both the accuracy and physicality of the long-term prediction and has good potential for informing decisions about stress irrigation.

This paper is organized as follows. Section II provides background on existing approaches for modeling and prediction in the context of SWC prediction. Section III presents the SWC prediction problem, the proposed model, and the training framework. In Section IV, numerical experiments are presented to evaluate the proposed approach and compare it against standard alternatives. Finally, Section V presents a summary of the work.

II. Background

Classical Soil-Water Content Modeling. Estimating soil moisture content is more intricate than it initially seems due to the nonlinear nature of soil behavior. The linear equation of soil-water balance can be expressed as follows:

$$\Delta\theta = P + Q_{in} - Q_{out} - ET - R$$

where θ is SWC, P is precipitation, ET is evapotranspiration, and Q_{in} and Q_{out} are lateral inflow and outflow [6]. This equation represents the major factors affecting SWC change. The aforementioned linear soil-water balance equation cannot capture saturation effects. The soil-water nonlinear behavior is captured through the water retention curve given by the Van Genuchten model [1]

$$\theta = \theta_r + \frac{\theta_s - \theta_r}{[1 + (\alpha \times h)^n]^m}$$

relating θ to pressure head h [2] in a nonlinear fashion. Here, θ_s and θ_r denote saturated and residual SWC values and α, n, m are empirical constants. All the model parameters depend on the soil type. Taking into account the soil-water nonlinear behavior allows us to extend the water balance equation to the Richards equation for water flow:

$$\frac{\partial\theta(h)}{\partial t} = \frac{\partial}{\partial z}[K(h)(\frac{\partial h}{\partial z} + 1)] - U$$

where U is root water uptake, z is vertical coordinate positive upward and $K(h)$ is the hydraulic conductivity [1] given by

$$K(h) = \frac{\{1 - (\alpha h)^{n-1}[1 + (\alpha h)^n]^{-m}\}^2}{[1 + (\alpha h)^n]^{m/2}}.$$

In two- or three-dimensional modeling, spatial extent is integrated into the non-linear soil-water modeling [3] and complex numerical simulation systems (e.g., HYDRUS-1D) are used to compute moisture values as a function of spatial location for different scenarios. To accurately simulate SWC values, all of the model parameters (e.g., soil density and type and root system presence) at each voxel must be available.

Deep learning approaches for SWC forecasting. Data-driven approaches, such as state-space models, have been considered in the realm of time series forecasting. These models operate by learning parameters from data, enabling the prediction of the next time step state based on the current state and input. By applying the state-space model repeatedly in an autoregressive manner, long-term forecasting can be achieved. This category includes nonlinear state space models like RNN, LSTM [4], or GRUs. Some recent work explores different training methods for time-series models like LSTM, combining teacher-forcing and autoregressive approaches to enhance long-term prediction accuracy (e.g., see [7] and [8]). Although the aforementioned can be used to formulate a solution for the SWC prediction problem in crop irrigation, only one work on this topic has been presented to our knowledge.

The study [5] employs LSTM and Bi-LSTM models to forecast soil moisture levels using soil and weather data, aiming to prevent drought stress in crops through smart irrigation. Both models accurately predict soil moisture up to 12 hours ahead. While some research addresses daily SWC prediction across large fields, solutions for plant-specific settings remain limited. In [9], the authors introduce an LSTM model with attention mechanisms to capture soil moisture and temperature patterns across ten flux tower sites. In [10], thirteen machine learning methods are evaluated to identify the best fit for SWC prediction using data from ten sites, while in [11] Conv-LSTM is applied to assess the impact of input features on SWC forecasting, including 2D spatial data, via explainable AI. These papers do not address directly the long-term prediction problem, though [9] and [11] consider it in their evaluation.

III. Problem Formulation and Approach

Notations: scalars are represented via small letters (e.g., t), vectors are represented using boldface small letters (e.g., $\mathbf{x}$, $\boldsymbol{\theta}$), and matrices are given by boldface capital letters (e.g., $\mathbf{M}$). Note that we use the convention that all vectors introduced are column vectors unless otherwise indicated. Similarly, scalar functions are represented using small letters and vector functions are represented via boldface small letters.

A. Problem Formulation

The problem addressed in this paper is the prediction of SWC values at future time instances based on (i) the

history of SWC, weather information described by evapotranspiration (see [6] for detail), and irrigation, and (ii) the future intended irrigation and forecasted ET_0. The problem formulation targets sequences that include SWC information from K different depth levels in the soil along with irrigation (Irr) details and reference evapotranspiration (ET_0) data. We represent SWC times-series at different depths as $SWC_1(t), \ldots, SWC_K(t)$. We denote the input vector time-series by $\mathbf{x}(t)$, for $t \in \{1, 2, \ldots, T\}$, where T is the sequence length, $\mathbf{x}(t) = [\mathbf{x}_1(t)^T, \mathbf{x}_2(t)^T]^T$ and

$$\mathbf{x}_1(t) = \begin{bmatrix} SWC_1(t) \\ SWC_2(t) \\ \vdots \\ SWC_K(t) \end{bmatrix}, \quad \mathbf{x}_2(t) = \begin{bmatrix} Irr(t) \\ ET_0(t) \end{bmatrix}. \tag{1}$$

Our prediction problem can be phrased as follows. Given the historical data up to time t, $\mathbf{x}(1), \ldots, \mathbf{x}(t)$ and a forecast $\mathbf{x}_2(t+1), \ldots, \mathbf{x}_2(t+T)$ of ET_0 and irrigation, our goal is to predict $\mathbf{x}_1(t+T)$ (or $\mathbf{x}_1(t+1), \ldots, \mathbf{x}_1(t+T)$).

B. Proposed Approach

To address the forecasting problem, we consider two key steps in our solution: (1) learning a nonlinear state-space model, which allows us to predict $\mathbf{x}(t+1)$ from the previous state $\boldsymbol{\theta}(t)$ and the current input $\mathbf{x}(t)$ and (2) recursively applying the one-step model for multiple steps to obtain $\mathbf{x}(t+1), \ldots, \mathbf{x}(t+W)$, where W is window size.

While in principle, our approach allows substituting different alternatives for the state-space model, we consider using recurrent neural network models. Such models maintain a hidden state to capture temporal dependencies denoted by $\boldsymbol{\theta}(t)$ and at each time step use the current observation to update the hidden state and predicted output. Specifically, given an initial hidden state $\boldsymbol{\theta}(0)$ and an input sequence $\mathbf{x}(1), \ldots, \mathbf{x}(T)$, the hidden state $\boldsymbol{\theta}(t)$ and the output $\mathbf{y}(t)$ for $t = 1, \ldots, T$ are obtained via the following state-space equations:

$$\begin{aligned} \boldsymbol{\theta}(t) &= \mathbf{g}(\mathbf{x}(t), \boldsymbol{\theta}(t-1)) \\ \mathbf{y}(t) &= \mathbf{f}(\boldsymbol{\theta}(t)). \end{aligned} \tag{2}$$

Note that this model can be applied recursively starting at an arbitrary t_0 to obtain $\mathbf{y}(t)$ as a function of: $\mathbf{x}(t_0+1), \ldots, \mathbf{x}(t)$ and $\boldsymbol{\theta}(t_0)$, which can be denoted by $y(t; \mathbf{x}(t_0+1), \ldots, \mathbf{x}(t), \boldsymbol{\theta}(t_0))$. This formulation holds for several well-known neural nets including RNNs such as LSTMs and GRUs (see [12] and [13]). Each of these models is characterized by a model-specific system function $\mathbf{g}$ and output function $\mathbf{f}$.

1) Training: We train the model based on fully labeled data sequences of the form $\mathbf{x}(1), \ldots, \mathbf{x}(T)$ where the target label for time t is $\mathbf{y}(t) = \mathbf{x}_1(t+1)$. That is, the targets are one-step-ahead predictions of SWC given information up to the current time. Our training strategy is designed to address two key challenges: (1) accurate long-term prediction, which is known to be challenging due to error propagation of one-step predictions [14], and (2) learning a model that satisfies basic physical constraints of water flow.

These challenges are addressed by a training loss consisting of three loss terms. The first and second loss terms are the Huber loss between the one-step ahead SWC prediction $\mathbf{y}(t)$ and its true value $\mathbf{x}_1(t+1)$ summed over a range of t values. For the first loss term, the one-step ahead prediction $\mathbf{y}(t)$ is obtained by forward propagation of the non-linear state-space model using ground truth data as input. In contrast, for the second loss term, the output $\mathbf{y}(t)$ is generated in an autoregressive manner, where SWC predictions at one-time step are fed as input to the model for the next time step, instead of using ground truth data. The third loss term is a regularization term used to control physical property violations by penalizing the number of instances where SWC is expected to decrease but the model predicts an increase. We proceed with a detailed description of the three loss terms used during training.

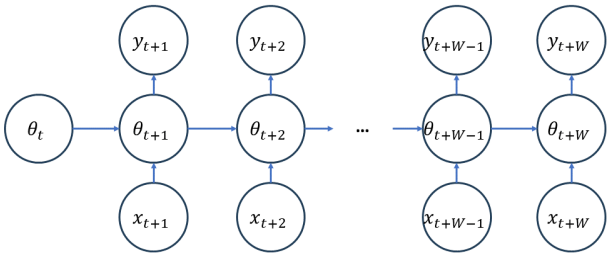

Fig. 2. Teacher Forcing training. Ground truth data is fed to the input layer. The initial state is obtained from the final state of the previous window.

Algorithm 1 Forward Function for TF or AR

Given $\theta(t), \mathbf{y}(t), \mathbf{x}(t+1), \mathbf{x}(t+2) \ldots \mathbf{x}(t+W)$ and type
for $i = t+1, t+2 \ldots t+W$ **do**
  if type = Teacher Forcing **then**
    $\boldsymbol{\theta}(i) = \mathbf{g}(\mathbf{x}(i), \boldsymbol{\theta}(i-1))$
  else if type = Autoregressive **then**
    $\boldsymbol{\theta}(i) = \mathbf{g}([\mathbf{y}(i-1)^T, \mathbf{x}_2(i)^T]^T, \boldsymbol{\theta}(i-1))$
  end if
  $\mathbf{y}(i) = \mathbf{f}(\boldsymbol{\theta}(i))$
end for
return $\mathbf{y}(t+1), \mathbf{y}(t+2) \ldots \mathbf{y}(t+W)$

2) First loss term: Teacher-Forcing (TF): This loss term is based on the teacher-forcing approach, providing sequential ground-truth data as input to the non-linear state-space model for one-time-step ahead predictions. The diagram in Fig. 2 shows the graphical representation of the one-step-ahead prediction model used in the teacher-forcing loss term during training. Ground truth data $[\mathbf{x}_{t+1}, \ldots, \mathbf{x}_{t+W}]$ is fed to the model during training. Algorithm 1 (with the TF option) describes how the model produces outputs $[\mathbf{y}(t+1), \ldots, \mathbf{y}(t+W)]$ from inputs $[\mathbf{x}(t+1), \ldots, \mathbf{x}(t+W)]$ and $\boldsymbol{\theta}(t)$.

During training, the following loss term is considered:

$$l_1(\mathbf{f}, \mathbf{g}) = \sum_{t=1}^{T-W-1} \sum_{t'=t+1}^{t+W} \frac{L_\delta(\mathbf{x}_1(t'+1) - \mathbf{y}(t'; t))}{(T-W-1)W} \tag{3}$$

where $\mathbf{y}(t';t)$ is used as a short-hand notation for $\mathbf{y}(t';\mathbf{x}(t+1),\ldots,\mathbf{x}(t'),\boldsymbol{\theta}(t))$ and $L_\delta(\cdot)$ denotes the Huber loss. The Huber loss associated with error $a \in \mathbb{R}$ is

$$L_\delta(a) = \begin{cases} \frac{1}{2}a^2 & \text{for } |a| \le \delta \\ \delta(|a| - \frac{1}{2}\delta) & \text{for } |a| > \delta \end{cases}$$

where $\delta \in \mathbb{R}^{++}$ is a threshold parameter determining the switch point between a quadratic loss and an absolute error loss. The loss applied to an error vector $\mathbf{a} = [a_1,\ldots,a_n]^T$ is applied element-wise, i.e., $L_\delta(\mathbf{a}) = \sum_{i=1}^n L_\delta(a_i)$. In (3), for each t in the outer summation, $\mathbf{y}(t+1;t),\ldots,\mathbf{y}(t+W;t)$ are computed as explained after (2) or equivalently in Algorithm 1 to form the objective in (3). Backpropagation through time (BPTT) for a long sequence may result in the vanishing gradients problem. This can be avoided, by restricting W, the length of a fixed-size window.

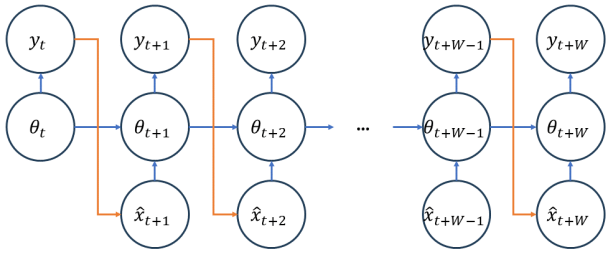

Fig. 3. Autoregressive Model Training: Ground truth data is fed only to the first input. Each unit's output is the next unit's input. The initial state is derived from the final state of the previous window.

3) Second loss term: Autoregressive Training (AR): The second loss term is based on an autoregressive approach, i.e., the model is trained to predict the future using its own predictions as inputs, as shown in Fig. 3 and as detailed in Algorithm 1 (using the AR option). Assuming training data is available up to time t, the SWC entries of the input at future times $t+1,\ldots,t+W$ are obtained from the outputs of the model as follows:

$$\hat{\mathbf{x}}_1(t+k) = \mathbf{y}(t+k-1), \qquad k = 1,2,\ldots,W. \tag{4}$$

This is illustrated in Fig. 3, where $\hat{\mathbf{x}}(t') = [\hat{\mathbf{x}}_1(t')^T, \mathbf{x}_2(t')^T]^T$ for $t' = t+1,\ldots,t+W$ is used as the input for prediction instead of $\mathbf{x}(t')$ (over the same range). Specifically, the orange arrows indicate that model predictions are fed to the next time step input. The following loss term is used to facilitate AR training:

$$l_2(\mathbf{f},\mathbf{g}) = \sum_{t=1}^{T-W-1} \sum_{t'=t+1}^{t+W} \frac{L_\delta(\mathbf{x}_1(t'+1) - \mathbf{y}(t';t))}{(T-W-1)W} \tag{5}$$

where $\mathbf{y}(t';t)$ here is used as a short-hand notation for $\mathbf{y}(t';\hat{\mathbf{x}}(t+1),\ldots,\hat{\mathbf{x}}(t'),\boldsymbol{\theta}(t))$. Specifically, for each t in the outer summation, $\mathbf{y}(t+1;t),\ldots,\mathbf{y}(t+W;t)$ are computed as explained in Algorithm 1 (using the AR option) to form the objective in (5). This term evaluates the error propagation due to repeated application of the model in an auto-regressive manner. By controlling this error, the training ensures that the model can be reliably applied multiple times to offer accurate long-term predictions. Probabilistic approaches have been previously proposed to combine AR and TF training [7]. We have found that this straightforward approach of combining AR and TF training allows for rapid convergence due to the TF term while significantly improving long-term forecasts compared to pure TF.

4) Third loss term: Regularization: One objective of this paper, aside from accurate long-term forecasts, is to ensure the model exhibits behavior similar to real soil, enhancing its credibility among growers. This is crucial when forecasting SWC in response to new irrigation patterns. We aim for the predicted changes in SWC, due to irrigation and ET_0 patterns, to be qualitatively consistent with historical data. We train the model on random irrigation patterns and regularize it based on predicted SWC changes between time steps to achieve this. We explore two regularization methods: one considering only the irrigation input, and another combining irrigation and ET_0.

5) Irrigation Regularization: For the purpose of obtaining the irrigation regularization loss, we generate a random future binary irrigation sequence: $Irr(t+1),\ldots,Irr(t+W)$ as part of $\mathbf{x}_2(t+1),\ldots,\mathbf{x}_2(t+W)$. Using the model in Fig. 3, we obtain the predicted value of the soil water content $\hat{\mathbf{x}}_1(t+1),\ldots,\hat{\mathbf{x}}_1(t+W)$. We then examine the change in the first entry of the SWC values in response to irrigation considering the following W examples: $(dSWC(t'), Irr(t'))$ for $t' = t+1,\ldots,t+W-1$, where $dSWC(t') = (\hat{\mathbf{x}}_1)_1(t'+1) - (\hat{\mathbf{x}}_1)_1(t')$. For each example, we aim to satisfy the qualitative constraint that SWC should only increase when irrigation is on, noting that there may be either a decrease or no change when irrigation is off. To promote this behavior, the penalty accumulates SWC changes occurring during an increase in predicted SWC when no irrigation is provided:

$$l_3(\mathbf{f},\mathbf{g}) = \frac{\lambda}{W-1} \sum_{t'=t+1}^{t+W-1} \big(dSWC(t')(1 - Irr(t'))\big)_+$$

where λ is the regularization parameter and $(x)_+ = \max\{0,x\}$. Note that this loss can be averaged over multiple random irrigation sequences.

6) Irrigation and ET_0 Regularization: In addition to irrigation, ET_0 also influences whether SWC increases or decreases. Here, we aim to regularize the model to capture the qualitative behavior of SWC changes with respect to irrigation and ET_0. For this purpose, we first train a simple classifier, based on historical data, for predicting the sign of SWC changes based on the irrigation and ET_0 data. In particular, the classifier uses the following convolutive form:

$$\sum_{i=0}^{W-1} (h_{ET}(i)ET_0(t-i) + h_{Irr}(i)Irr(t-i)) + \beta. \tag{6}$$

Since this predictor uses unknown coefficients, the coefficients $\mathbf{h}_{ET}$, $\mathbf{h}_{Irr}$, and β are trained to minimize the following loss:

$$\sum_{t'=t+1}^{t+W} \ell_{CE}(\mathbf{h}_{ET}^T \mathbf{et}_0(t') + \mathbf{h}_{Irr}^T \mathbf{irr}(t') + \beta, dSWC(t')) \tag{7}$$

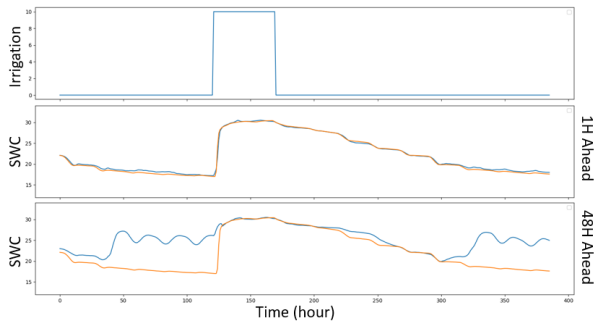

Fig. 4. SWC Prediction Comparison for TF Algorithm - Cabernet Sauvignon 94w (Orange curve: Ground Truth, Blue curve: Prediction)

where

$$\begin{aligned}\mathbf{h}_{ET} &= [h_{ET}(0),\ldots,h_{ET}(W-1)]^T,\\ \mathbf{h}_{Irr} &= [h_{Irr}(0),\ldots,h_{Irr}(W-1)]^T,\\ \mathbf{et}_0(t') &= [ET_0(t'),\ldots,ET_0(t'-W+1)]^T,\\ \mathbf{irr}(t') &= [Irr(t'),\ldots,Irr(t'-W+1)]^T,\end{aligned}$$

and $\ell_{CE}(\cdot,\cdot)$ is the cross-entropy loss given by $\ell_{CE}(s,y) = -ys + \log(1+e^s)$. The goal is to optimize the model while penalizing sign disagreement between the predictor of SWC change in (6) with the change of SWC due to the trained model. Therefore, the following loss term

$$l_3(\mathbf{f},\mathbf{g}) = \frac{\lambda}{W-1}\sum_{t'=t+1}^{t+W-1}\big(\mathbf{I}(\mathbf{h}^*_{ET}\cdot\mathbf{et}_0(t')+ \mathbf{h}^*_{Irr}\cdot\tilde{\mathbf{irr}}(t')+\beta^*<\alpha)\cdot(dSWC(t')-A)\big)_+ \quad (8)$$

is used as regularization during training. In the mathematical expressions above, $\tilde{\mathbf{irr}}$ denotes the randomized irrigation, and both α and A are used to define a margin near the origin, capturing approximately 2% of all data points situated in the second quadrant for penalization during the training phase. As a result, the model incurs a penalty when the conditions $\mathbf{h}^*_{ET}\cdot\mathbf{et}_0(t') + \mathbf{h}^*_{Irr}\cdot\tilde{\mathbf{irr}}(t') + \beta^* < \alpha$ and $dSWC > A$ are met, and the penalty is proportional to $(dSWC - A)$. Using this penalty term, models that produce a SWC increase when (6) predicts a SWC decrease, are discouraged.

IV. Experiments

A. Dataset

The time-series dataset was recorded at the Ciel du Cheval vineyard located in Eastern Washington in 2017. The dataset consists of hourly SWC recordings collected at six specific depths under the ground at 8, 16, 24, 32, 40, and 48 inches; irrigation information; and ET_0 information from weather forecast stations. The data used in this work is from 5 different sites, denoted by: Cabernet Franc 94, Cabernet Franc 98, Cabernet Sauvignon 98, Cabernet Sauvignon 94e, and Cabernet Sauvignon 94w, which specify both the plant cultivar and numeric field block code. The duration of the data from each site differs: 3 months for Cabernet Franc 94, 4 months for Cabernet Franc 98, 5 months for Cabernet Sauvignon 98, and 6.5 months for both Cabernet Sauvignon 94e and 94w irrigation blocks.

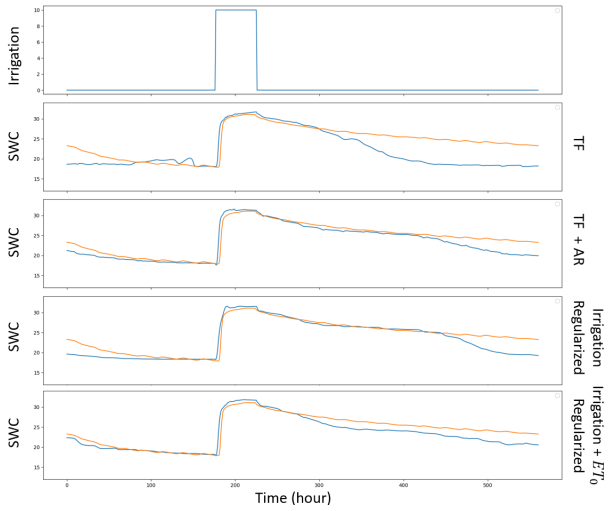

Fig. 5. SWC Prediction Algorithm Comparison (48 Hours Ahead) - Cabernet Sauvignon 94e (Orange curve: Ground Truth, Blue curve: Prediction)

B. Algorithms compared

In our study, we trained LSTM models using four training algorithms: TF; TF and AR; TF, AR, and Irrigation regularization; and TF, AR, and Irrigation-ET_0 regularization. We also used classical ARX (autoregressive with exogenous variables) models with and without stability constraints as a baseline to compare our models with a well-known forecasting method.[1]

In our experiments, we use single-layer LSTM models with an input size of 8, an output size of 6, and a hidden state dimension of 128. The LSTM cell parameters are updated over 5000 epochs, with the dataset processed in batches of 256. Overlapping sequences are formed with a window size of $W = 48$, sliding by one step per sequence. To address regularization, we generate fictitious irrigation sequences based on the statistical characteristics of observed on/off periods in the irrigation data. The 'Adam' algorithm is used for optimization.

1) TF: State-space models are trained with only the TF loss as stated in the previous section.

2) TF+AR: This model is trained using both the TF and AR loss terms. The TF loss enforces one-step-ahead training, while the AR loss allows accurate long-term predictions by minimizing error-propagation.

3) TF+AR+Regularization: We evaluate the models trained with TF+AR along with the regularization terms. Incorporating regularization terms enhances the physicality of the predictions, resulting in more reliable models.

[1]In a preliminary consideration of [9] as a candidate algorithm for comparison, we have found that a naive predictor that uses the last sample and a linear autoregressive model outperform their method on the dataset used in their paper. Hence, we elected to compare directly to an ARX model in place of the model in [9].

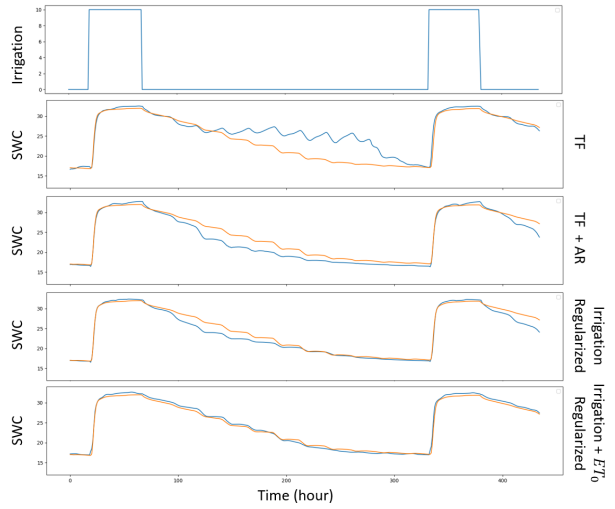

Fig. 6. SWC Prediction Algorithm Comparison (48 Hours Ahead) - Cabernet Franc 98 (Orange curve: Ground Truth, Blue curve: Prediction)

4) Baseline: ARX models are used in biological system modeling [15], which is structurally similar to our problem, and have also been applied to soil moisture data. Bwambale et al. modeled soil moisture dynamics for smart irrigation scheduling with various methods, including ARX [16]. ARX processes are linear difference equations with constant coefficients, representing stationary time series. We use the ARX model, to define the relation between the current SWC vector, its past values, and the current/past values of ET_0 and irrigation:

$$\mathbf{x}_1(t) = \sum_{l=1}^{p} \mathbf{A}_l \mathbf{x}_1(t-l) + \sum_{l=0}^{q} \mathbf{B}_l \mathbf{x}_2(t-l) + \mathbf{e}(t)$$

where $\mathbf{e}(t)$ are $i.i.d.$ distributed $\mathcal{N}(\mathbf{0}_K, \mathbf{I}_K)$. The matrices $\mathbf{A}_l$ and $\mathbf{B}_l$ are the model parameters that relate the SWC vector to its past values and the current and past values of irrigation/ET_0. Parameters can be estimated using maximum likelihood estimation [17]:

$$\min_{\gamma} \|\mathbf{x}_1(t) - (\sum_{l=1}^{p} \mathbf{A}_l \mathbf{x}_1(t-l) + \sum_{l=0}^{q} \mathbf{B}_l \mathbf{x}_2(t-l))\|^2$$

where $\gamma = \{\mathbf{A}_1, \ldots, \mathbf{A}_p, \mathbf{B}_0, \ldots, \mathbf{B}_q\}$. Stability constraints are important for long-term forecasts; without them, forecasts may diverge [18]. Stability is ensured by appropriately restricting the search space over $\{\mathbf{A}_1, \ldots, \mathbf{A}_p\}$ (see [17] and [18]). There are various methods to ensure stability, such as using Levinson mapping and geometry [19] or finding the closest stable solution from an unstable approximation [20]. Both stability-constrained (ARX-S) and -unconstrained (ARX) models are implemented to compare with the proposed methods. Note that only for ARX-S models, matrices $\mathbf{A}_l$'s were assumed diagonal to simplify the implementation.

C. Evaluation Metrics

To evaluate each algorithm, we consider three performance metrics applied to the test data. Root mean squared error (RMSE) is used to evaluate the prediction accuracy:

$$RMSE = \sqrt{\frac{1}{T'-T} \sum_{t=T+1}^{T'} \|\mathbf{x}_1(t+W+1) - \mathbf{y}(t+W;t)\|^2}$$

where the limits of the summation $T+1$ and T' represent the beginning and end time of the evaluation period. We selected this metric to maintain consistency in units with SWC values. Additionally, two sign disagreement metrics are introduced to quantify the instances when the direction of change in SWC predicted by the model disagrees with the expected direction of change given by the predictor. We introduce a random irrigation input (using similar statistics to the real irrigation data) and count the fraction of time the model-predicted SWC sign change disagrees with the sign of the predictor. To capture the differences in the two regularization methods, i.e., one conveys both irrigation and ET_0 information while the other one only conveys the irrigation information, we use two sign disagreement metrics. The first metric

$$SD_1 = \frac{|\{t' | (1 - Irr(t')) \cdot dSWC(t') > 0, t' \in \mathcal{S}\}|}{W}$$

counts the number of instances in which no irrigation produces an increase in SWC. The second is

$$SD_2 = \frac{|\{t' | d\hat{S}WC(t') < \alpha, dSWC(t') > A, t' \in \mathcal{S}\}|}{W}$$

where α and A are defined in section III-B6, $\mathcal{S} = [t+1, \ldots, t+W]$, and $d\hat{S}WC$ is defined as

$$d\hat{S}WC(t') = (\mathbf{h}^*_{ET} \cdot \mathbf{et}_0(t') + \mathbf{h}^*_{Irr} \cdot \mathbf{irr}(t') + \beta^*).$$

This metric counts the instances predicted to have negative SWC change while introducing positive SWC change.

D. Data stratification

Since the dataset covers only one dynamically changing growing season, we employed a specific approach to training and testing splits. We allocated 85% of the time-series data for training and 15% for testing, with the last 15% of the training set used for validation. Moreover, we considered prediction performance at multiple time points by constructing multiple windows of increasing size. We extended the data in increments, starting with a base length of $W_1 = 2100$ data points. We then trained separate models with partially overlapping sequences ending at W_1, $W_1 + W_2$, $W_1 + 2W_2$, $W_1 + 3W_2$, and $W_1 + 4W_2$ (the full length), where W_2 is the increment step size. This approach reduces the likelihood that outliers in the data would dominate the performance.

E. Hyper-parameter tuning

For the methods that employ regularization, we consider hyper-parameter tuning to determine the optimal value of λ. For each window, we train a model using $\lambda \in \{0.1, 1, 10\}$ and among the three values select λ which minimizes the model comparison metric, SD$\times$RMSE, across five windows. This approach balances model accuracy with physicality, offering a trade-off between the two since using regularization yields

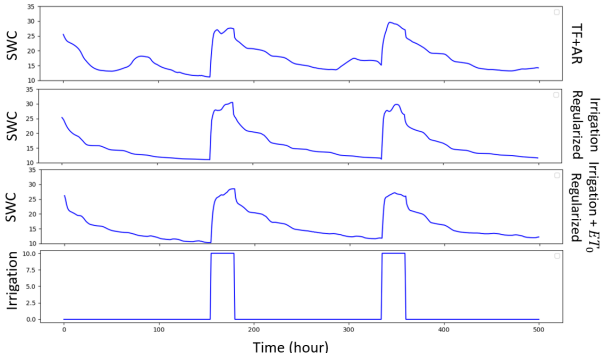

Fig. 7. SWC Forecast Comparison with Fictitious Irrigation - Cabernet Franc 94 - LSTM Model

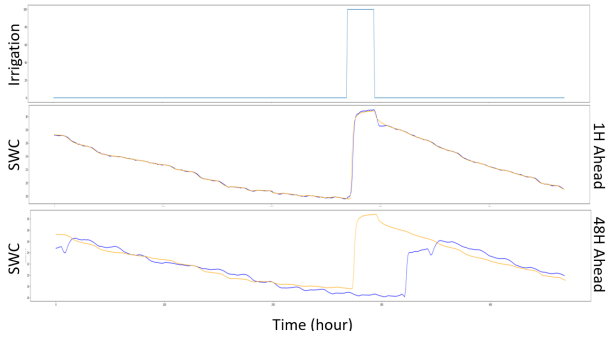

Fig. 8. SWC Prediction Comparison for ARX-S Algorithm - Cabernet Sauvignon 98 (Orange curve: Ground Truth, Blue curve: Prediction)

to compromise between the two. To reduce variation due to initialization, we initialize the model with the one obtained from the previous window. We then train the model for 2000 epochs, ensuring fair comparisons among models with the same λ value and training type. This is done only if the considered window's neighbors have significantly less RMSE.

F. Results

1) Qualitative Results: As observed in Fig. 4, TF-trained models excel in one-step ahead predictions but struggle to predict SWC multiple steps ahead. Figures 5 and 6 show that both TF and TF+AR models perform well in one-step ahead predictions. The TF+AR model outperforms the TF model in long-term SWC predictions by learning relationships more effectively and reducing error accumulation, while the TF model suffers from error propagation due to non-autoregressive training. Despite this, Figure 7 shows that without regularization the TF+AR model may incorrectly anticipate irrigation response when irrigation is absent, suggesting a potential violation of the physical behavior of the model.

Figures 5 and 6 show that the proposed regularized models can predict SWC values 48 hours ahead of the given data, demonstrating a close fit between ground truth and predictions compared to non-regularized models. Although regularization introduces a slight drop in precision, such a drop is minor. Our regularization methods seem to improve accuracy relative to TF and TF+AR models in some cases and enforce the model's physical behavior. Different regularization methods alter results slightly but still achieve the desired outcome, with the Irrigation + ET_0 regularization fitting ground truth SWC better. Additionally, the baseline ARX model can make 1-hour ahead predictions but fails to capture SWC behavior in 48-hour predictions, highlighting its limitations in capturing non-linear behavior and causality, as shown in Fig. 8.

2) Quantitative Results: Quantitative results for models across various irrigation blocks, sequence extensions, and algorithms are presented in Table I. Each irrigation block, with a distinct plant and irrigation strategy, exhibits varied behaviors in RMSE and sign disagreements. The table highlights the best RMSE and sign disagreement values in bold.

Generally, TF models have higher RMSE than TF+AR models due to error propagation in long-term predictions, with some exceptions observed in Table I. It is also demonstrated that irrigation-regularized models consistently show lower SD_1 values, except in two cases, and often lower or comparable RMSE to TF+AR models. Additionally, irrigation and ET_0 regularized models mostly achieve reduced SD_2 values, except in five cases, while maintaining lower or comparable RMSE values. This confirms the success of our regularization methods in capturing soil physics behavior and aligns with the primary objective of minimizing sign disagreements while achieving a good fit for the dataset. Additionally, the baseline ARX and ARX-S models show higher RMSE values than other models, indicating the superiority of non-linear models in predicting SWC dynamics over the linear ARX model.

Finally, it is crucial to note that the regularization addresses rare real-world phenomena, quantified in Table II. The SD_1 and SD_2 values in this table serve as benchmarks for ensuring the physical realism and reliability of the forecasts.

V. Conclusion

This paper addresses the challenges of training models for long-term SWC prediction in grapevines. We demonstrated that a simple three-term loss function enables non-linear state-space models, like LSTM, to avoid error propagation while satisfying physical constraints. Evaluations on months of hourly real-world data from five test sites show that our models can aid deficit irrigation decisions. Future work could explore combining these methods with multi-task learning, treating each sensor site as a distinct task with unique physical characteristics, to enhance overall prediction accuracy.

References

[1] 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.

[2] R. F. Carsel and R. S. Parrish, "Developing joint probability distributions of soil water retention characteristics," *Water resources research*, vol. 24, no. 5, pp. 755–769, 1988.

[3] J. Šimůnek and M. T. van Genuchten, "Modeling nonequilibrium flow and transport processes using HYDRUS," *Vadose Zone Journal*, vol. 7, no. 2, pp. 782–797, 2008.

[4] M. Schüssler, T. Münker, and O. Nelles, "Deep recurrent neural networks for nonlinear system identification," in *IEEE Symposium Series on Computational Intelligence*, 2019, pp. 448–454.

TABLE I
RMSE AND SIGN DISAGREEMENTS FOR EACH ALGORITHM BY IRRIGATION BLOCK AND EXTENSIONS (EXT.)

		Cab. Franc 94		Cab. Franc 98		Cab. Sauvignon 94e		Cab. Sauvignon 94w		Cab. Sauvignon 98	
	Algorithm	RMSE	SD_1 / SD_2	RMSE	SD_1 / SD_2	RMSE	SD_1 / SD_2	RMSE	SD_1 / SD_2	RMSE	SD_1 / SD_2
Ext. 0	TF	**0.943**	0.181 / 0.116	0.919	0.182 / 0.244	**1.228**	0.138 / 0.077	**1.270**	0.105 / 0.075	2.470	0.258 / 0.182
	TF+AR	1.603	0.173 / 0.135	**0.707**	0.172 / 0.215	1.448	0.148 / 0.071	1.754	0.098 / 0.079	**2.15**8	**0.257** / 0.133
	TF+AR+Irr Reg.	1.060	**0.006** / 0.059	0.896	**0.007** / 0.196	1.597	**0.032** / 0.043	1.471	**0.028** / 0.056	2.314	0.265 / 0.146
	TF+AR+Irr+ET_0 Reg.	1.090	0.066 / **0.040**	0.815	0.154 / **0.108**	1.456	0.148 / **0.033**	1.683	0.090 / **0.055**	2.719	0.282 / **0.093**
	ARX-S	5.700	0.392 / 0.332	5.834	0.281 / 0.244	3.816	0.200 / 0.153	3.674	0.181 / 0.102	5.981	0.395 / 0.267
	ARX	7.561	0.373 / 0.347	5.413	0.301 / 0.283	12.890	0.425 / 0.374	5.045	0.396 / 0.297	5.206	0.391 / 0.341
Ext. 1	TF	3.907	0.334 / 0.242	0.586	0.121 / 0.197	1.596	0.176 / 0.040	1.118	0.168 / 0.081	**1.966**	0.186 / 0.113
	TF+AR	1.312	0.152 / 0.105	0.509	0.127 / **0.105**	0.989	0.188 / 0.045	1.208	0.233 / 0.084	1.968	0.184 / 0.103
	TF+AR+Irr Reg.	0.963	0.099 / 0.068	**0.507**	**0.038** / 0.182	**0.945**	**0.066** / 0.032	**0.792**	**0.029** / 0.064	2.297	**0.026** / 0.094
	TF+AR+Irr+ET_0 Reg.	**0.841**	**0.063** / **0.027**	0.544	0.163 / 0.123	1.066	0.177 / **0.010**	1.069	0.061 / **0.041**	2.394	0.132 / **0.069**
	ARX-S	6.036	0.370 / 0.306	4.947	0.296 / 0.250	3.883	0.322 / 0.205	3.558	0.256 / 0.093	4.061	0.384 / 0.226
	ARX	7.331	0.432 / 0.404	4.881	0.340 / 0.317	4.099	0.418 / 0.391	3.806	0.381 / 0.345	6.086	0.482 / 0.421
Ext. 2	TF	1.706	0.253 / 0.183	1.073	0.059 / 0.204	2.746	0.399 / 0.115	2.177	0.304 / 0.153	**1.141**	0.167 / 0.129
	TF+AR	0.860	0.183 / 0.078	0.401	0.082 / 0.078	1.157	0.259 / 0.107	0.669	0.211 / 0.109	1.152	0.229 / 0.110
	TF+AR+Irr Reg.	0.990	**0.013** / **0.054**	0.349	**0.013** / 0.180	**0.789**	**0.140** / 0.066	**0.368**	**0.099** / **0.080**	1.208	**0.129** / 0.093
	TF+AR+Irr+ET_0 Reg.	**0.774**	0.146 / 0.059	**0.345**	0.118 / **0.077**	0.791	0.151 / **0.043**	0.510	0.185 / 0.107	1.324	0.283 / **0.080**
	ARX-S	6.046	0.395 / 0.295	3.923	0.275 / 0.240	3.537	0.173 / 0.083	3.226	0.202 / 0.083	4.415	0.214 / 0.118
	ARX	7.987	0.432 / 0.384	4.336	0.391 / 0.312	3.771	0.424 / 0.382	3.433	0.406 / 0.352	4.729	0.289 / 0.222
Ext. 3	TF	1.001	0.169 / 0.126	1.191	0.235 / 0.205	1.148	0.214 / 0.071	1.071	0.275 / 0.130	2.339	0.297 / 0.070
	TF+AR	0.930	0.137 / **0.052**	1.096	0.159 / **0.052**	1.121	0.080 / 0.064	0.853	0.147 / 0.108	1.099	0.131 / **0.070**
	TF+AR+Irr Reg.	0.909	**0.010** / 0.064	**0.658**	**0.011** / 0.156	1.012	**0.032** / **0.061**	**0.604**	**0.035** / 0.072	1.261	**0.026** / 0.080
	TF+AR+Irr+ET_0 Reg.	**0.791**	0.105 / 0.053	0.849	0.091 / 0.070	**0.657**	0.180 / 0.078	1.005	0.076 / **0.064**	**0.819**	0.138 / 0.076
	ARX-S	7.034	0.397 / 0.317	4.169	0.230 / 0.195	3.405	0.185 / 0.088	3.089	0.203 / 0.085	3.921	0.374 / 0.281
	ARX	8.238	0.435 / 0.406	4.335	0.326 / 0.323	3.746	0.431 / 0.395	3.047	0.421 / 0.361	3.958	0.394 / 0.275
Ext. 4	TF	1.305	0.227 / 0.145	1.777	0.316 / 0.227	2.239	0.142 / 0.026	1.225	0.075 / 0.065	1.335	0.115 / 0.096
	TF+AR	**0.860**	0.040 / 0.061	1.476	0.081 / **0.061**	1.047	0.107 / 0.027	0.725	0.151 / 0.070	**1.130**	0.178 / 0.104
	TF+AR+Irr Reg.	0.936	**0.008** / 0.069	0.929	**0.017** / 0.105	1.299	**0.057** / **0.019**	0.666	**0.064** / **0.060**	1.230	**0.048** / 0.093
	TF+AR+Irr+ET_0 Reg.	1.058	0.119 / **0.026**	**0.907**	0.120 / 0.149	**1.019**	0.085 / 0.020	**0.569**	0.141 / 0.069	1.405	0.129 / **0.044**
	ARX-S	6.288	0.370 / 0.272	4.136	0.241 / 0.197	3.030	0.378 / 0.242	2.697	0.350 / 0.147	2.203	0.214 / 0.118
	ARX	8.167	0.422 / 0.401	4.515	0.391 / 0.348	3.357	0.419 / 0.348	2.890	0.423 / 0.230	2.395	0.415 / 0.308

TABLE II
DATASET SIGN DISAGREEMENT VALUES FOR EACH CULTIVAR

	Cab. Franc 94	**Cab. Franc 98**	**Cab. Sauvignon 94e**	**Cab. Sauvignon 94w**	**Cab. Sauvignon 98**
$\mathbf{SD_1}$	0.053	0.099	0.096	0.120	0.115
$\mathbf{SD_2}$	0.0197	0.0193	0.0198	0.0192	0.0197

[5] P. Suebsombut, A. Sekhari, P. Sureephong, A. Belhi, and A. Bouras, "Field data forecasting using LSTM and Bi-LSTM approaches," *Applied Sciences*, vol. 11, no. 24, 2021.

[6] T. Wang, Z. Wu, P. Wang, T. Wu, Y. Zhang, J. Yin, J. Yu, H. Wang, X. Guan, H. Xu, D. Yan, and D. Yan, "Plant-groundwater interactions in drylands: A review of current research and future perspectives," *Agricultural and Forest Meteorology*, vol. 341, p. 109636, 2023.

[7] S. Bengio, O. Vinyals, N. Jaitly, and N. Shazeer, "Scheduled sampling for sequence prediction with recurrent neural networks," *Advances in neural information processing systems*, vol. 28, 2015.

[8] P. Teutsch and P. Mäder, "Flipped classroom: Effective teaching for time series forecasting," *Transactions on Machine Learning Research*, 2022.

[9] Q. Li, Y. Zhu, W. Shangguan, X. Wang, L. Li, and F. Yu, "An attention-aware LSTM model for soil moisture and soil temperature prediction," *Geoderma*, vol. 409, p. 115651, 2022.

[10] Y. Wang, L. Shi, Y. Hu, X. Hu, W. Song, and L. Wang, "A comprehensive study of deep learning for soil moisture prediction," *Hydrology and Earth System Sciences*, vol. 28, no. 4, pp. 917–943, 2024.

[11] F. Huang, Y. Zhang, Y. Zhang, W. Shangguan, Q. Li, L. Li, and S. Jiang, "Interpreting Conv-LSTM for spatio-temporal soil moisture prediction in China," *Agriculture*, vol. 13, no. 5, 2023.

[12] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[13] I. D. Jordan, P. A. Sokół, and I. M. Park, "Gated recurrent units viewed through the lens of continuous time dynamical systems," *Frontiers in Computational Neuroscience*, vol. 15, 2021.

[14] S. Ross, G. Gordon, and D. Bagnell, "A reduction of imitation learning and structured prediction to no-regret online learning," in *proc. Intl. Conf. Artificial Intelligence and Statistics*, vol. 15, 2011, pp. 627–635.

[15] M. H. Perrott and R. J. Cohen, "An efficient approach to ARMA modeling of biological systems with multiple inputs and delays," *IEEE Transactions on Biomedical Engineering*, vol. 43, no. 1, 1996.

[16] E. Bwambale, F. K. Abagale, and G. K. Anornu, "Data-driven modelling of soil moisture dynamics for smart irrigation scheduling," *Smart Agricultural Technology*, vol. 5, p. 100251, 2023.

[17] G. E. P. Box, G. M. Jenkins, G. C. Reinsel, and G. M. Ljung, *Time series analysis: forecasting and control*, 5th ed., ser. Wiley Series in Probability and Statistics. Wiley, 2016.

[18] P. J. Brockwell and R. A. Davis, "Introduction to time series and forecasting," Switzerland, 2016.

[19] P. Combettes and H. Trussell, "Best stable and invertible approximations for ARMA systems," *IEEE Transactions on Signal Processing*, vol. 40, no. 12, pp. 3066–3069, 1992.

[20] R. Moses and D. Liu, "Determining the closest stable polynomial to an unstable one," *IEEE Transactions on Signal Processing*, vol. 39, no. 4, pp. 901–906, 1991.