Expert Systems With Applications 207 (2022) 117653

Contents lists available at ScienceDirect

Expert Systems With Applications

journal homepage: www.elsevier.com/locate/eswa

Soil moisture forecast for smart irrigation: The primetime for machine learning

Rodrigo Togneri [a,b,*], Diego Felipe dos Santos [c], Glauber Camponogara [d], Hitoshi Nagano [a], Gilliard Custódio [b], Ronaldo Prati [b], Stênio Fernandes [e], Carlos Kamienski [b]

[a] Fundação Getúlio Vargas (FGV/EAESP), Rua Itapeva, 474 / 9th floor, São Paulo, SP 01332-000, Brazil
[b] Universidade Federal do ABC (UFABC), Avenida dos Estados, 5001 B / 10th floor, Santo André, SP 09210-580, Brazil
[c] Agrosmart, Rua Pereira Tangerino, 128, Campinas, SP 13073-210, Brazil
[d] Tempo OK – Meteorology, Rua M.M.D.C, 450 / 5th floor / room 501, São Paulo, SP 05510-020, Brazil
[e] ServiceNow, 6650 St. Urbain Street, Suite 500, Montreal, QC H2S 3G9, Canada

ARTICLE INFO

Keywords:
Smart irrigation
Machine learning
Soil moisture forecast
Water need estimation
Internet-of-Things

ABSTRACT

The rise of the Internet of Things allowed higher spatial–temporal resolution soil moisture data captured through in situ sensing. Such abundance of data enables machine learning-based soil moisture forecast as an alternative to traditional mechanistic approaches for irrigation water need estimation. This paper develops a guideline for soil moisture forecast modeling based on machine learning, tested in a real case analysis comprehending eight crop types in twelve fields from four farms distributed over diverse climatic scenarios in Brazil. Instead of a single value, we predict the following days' minimum and maximum values as targets to monitor risks of extreme soil moisture values. Furthermore, modeling soil moisture directly in volumetric water content (VWC) is better than modeling soil matric potential (SMP) to later convert in soil moisture VWC. We test several algorithms and find out that LightGBM outperforms linear regression, decision tree, random forest, multilayer perceptron, LSTM, and StemGNN. Also, blending predictions via algorithm ensemble provides an additional accuracy gain. For model training and accuracy measurement, we use weighted datasets to privilege rare but critical data points. We show that soil moisture forecast reaches its maximum performance considering only past soil moisture, a context-aware index, and a precipitation forecast. Finally, we demonstrate that traditional domain-knowledge features - such as evapotranspiration, crop phenology, and soil hydraulic behavior - are not relevant to improving SM forecast performance. Consequently, our paper suggests full data-driven approaches for irrigation water need estimation, observed some care regarding data quality.

1. Introduction

Precision Irrigation (PI) is the application of the right amount of water to the soil at the right time and in the right places to provide water-related crop productivity (Kamienski et al., 2019). Broadly, PI can keep high crop yields using minimal water, thus reducing the environmental footprint and production costs such as electricity needed for pumping water through irrigation systems (Abrishambaf, Faria, Gomes, & Vale, 2020; García, Montesinos, Poyato, & Díaz, 2016). Water is a scarce natural resource that must be preserved. In some regions, water is simply not available for all farms and crops simultaneously, making a case for using sophisticated optimization techniques (Shahdany, Maestre, & van Overloop, 2015). As water preservation is fundamental for the food safety of future generations and agriculture consumes 70–80% of the world's freshwater, it becomes evident that water use efficiency in agriculture also plays an essential socio-economical and environmental role (Grafton et al., 2018).

Soil moisture (SM) is the amount of water stored in the soil within the plant's root zone, being the primary source of water for plants (Jensen & Allen, 2016). An ideal range of SM for each soil type favors plant development, as plants absorb water readily (Rawls, Gish, & Brakensiek, 1991; Van Genuchten, 1980). PI relies on water need estimation (WNE),

* Corresponding author at: Fundação Getúlio Vargas (FGV/EAESP), Rua Itapeva, 474 / 9th floor, São Paulo, SP 01332-000, Brazil.
E-mail addresses: rodrigo.togneri@fgv.br (R. Togneri), diego@agrosmart.com.br (D. Felipe dos Santos), glauber.camponogara@tempook.com (G. Camponogara), hitoshi.nagano@fgv.br (H. Nagano), gilliard.custodio@ufabc.edu.br (G. Custódio), ronaldo.prati@ufabc.edu.br (R. Prati), stenio.fernandes@servicenow.com (S. Fernandes), carlos.kamienski@ufabc.edu.br (C. Kamienski).

https://doi.org/10.1016/j.eswa.2022.117653
Received 9 December 2021; Received in revised form 14 April 2022; Accepted 27 May 2022
Available online 31 May 2022

i.e., calculating the right amount of water applied to the soil, to keep SM within the ideal range until the next irrigation opportunity (Jensen & Allen, 2016).

Before the rise of the Internet of Things (IoT), soil sensing and field communication technologies were unreliable and costly. Measuring SM directly in the soil with high precision and a fine-grained spatial–temporal resolution was not feasible. As a result, WNE models used indirect approaches based on soil water balance: water outtake estimated mainly by evaporation and plant transpiration proxies; water intake primarily via precipitation; and where water need estimate is the calculated as the water deficit between water intake and outtake (Jensen & Allen, 2016). Traditional soil water balance approaches are still the current paradigm. Evaporation, plant transpiration, and precipitation are weather-dependent, and the weather forecast is currently a well-established discipline. On the other hand, WNE models now apply both realized and forecasted values, frequently using simulation as a tool to evaluate future scenarios (Domínguez-Niño, Oliver-Manera, Girona, & Casadesús, 2020; Gu et al., 2017; Jensen & Allen, 2016; Linker, Ioslovich, Sylaios, Plauborg, & Battilani, 2016).

As precision irrigation met IoT, allowing for more granular data capturing and higher automation levels, the so-called smart irrigation (SI) era started. With the rise of IoT and the consequent availability of improved and more accessible soil sensing and field communication technologies, direct SM measurements became feasible with higher precision and spatial–temporal resolution (García, Parra, Jimenez, Lloret, & Lorenz, 2020; Kamienski et al., 2019). With reliable SM data, SM monitoring and forecasting returned to the agenda, becoming an alternative to the soil water balance indirect estimates (Karandish & Šimůnek, 2016; Adeyemi, Grove, Peets, Domun, & Norton, 2018; Gumière et al., 2020; Yu, Zhang, Xu, Dong, & Zhangzhong, 2021; Dubois, Teytaud, & Verel, 2021; Ahmed, Deo, Raj, Ghahramani, Feng, Yin, & Yang, 2021; Wang, Yuan, Liu, Xing, Ji, Li, & Mo, 2022). The availability of a vast amount of data acknowledged machine learning (ML) as a promising approach to SM forecast (Karandish & Šimůnek, 2016; Adeyemi et al., 2018; Gumière et al., 2020; Yu et al., 2021; Dubois et al., 2021; Ahmed, et al., 2021; Wang, et al., 2022).

The general objective of this paper is to improve machine learning-based SM forecasting for smart irrigation applications. Although privileging a data-driven approach, we still evaluate the potential influence of traditional domain-knowledge factors, such as physical and empirical-based ET, soil, and crop features (Jensen & Allen, 2016), on SM forecast. To unfold the general objective, we establish the following research questions. (i) Target variable definition: Is it better to use extreme daily values (minimum and maximum) as the target variable instead of a single value collected at an arbitrary daytime? Or even instead of daily mean and variance? What is the best target variable between two mainstream ways to measure water in soil: relative volumetric water content or the difficulty accessing the plant's water? (ii) Context-awareness importance: Can a context-aware engineered feature improve ML-based SM forecast performance based on all sensed data? (iii) Feature group selection: What are the most important group of features and data sources to ML-based SM forecast? Are domain-knowledge features (as evaporation, transpiration, crop phenology, and soil models) important factors? (iv) Algorithm comparison: What is the best ML approach between the two classes considered state-of-the-art for time-series problems (Géron, 2019; Cao D., et al., 2020; Microsoft, 2021b; Borisov, Leemann, SeBler, Haug, Pawelczyk, & Kasneci, 2022): decision tree ensembles and recurrent neural networks (RNN)? (v) Algorithm ensemble utility: Does algorithm blending increase accuracy? (vi) Dataset weighting utility for training and measuring performance: What are the advantages and disadvantages of weighting the dataset to privilege critical data points (those that are harder to forecast)? What are the effects if we use this setup for model training and performance measurement?

These objectives are investigated by running ML-based SM forecast modeling for a real case based on data sourced by a Brazilian smart irrigation provider, Agrosmart (Agrosmart, 2021), covering mid-2016 to mid-2020 of twelve fields from four farms located in Brazil. The data samples are climatic and geographically diversified. The climates covered correspond to 25.5% of the world's area suitable for agriculture (section 4), and consider eight different crop types, advocating that our findings may generalize well within this domain.

In the remainder of this paper: section 2 gives the background; section 3 brings related work; section 4 describes our real case and their data; section 5 details our methodology; section 6 presents and discusses our results; section 7 concludes this work, summarizing our findings and, finally, section 8 points out next steps.

2. Background

This section aims to provide the necessary knowledge about soil water measurement, soil water dynamics, the relation between soil moisture and weather conditions, traditional and data-driven approaches to water need estimation, and fundamentals of the ML techniques that will be used here.

2.1. Soil water measurement

Two main concepts for measuring soil water in agronomy are soil moisture and soil matric potential (SMP). Also known as soil water content, soil moisture (SM) is the amount of water stored in the soil within the plant's root zone. It is usually measured in volumetric water content (VWC), defined as the ratio of water volume to soil volume, which, in turn, is commonly represented in $[cm^3 \bullet cm^{-3}]$ or simply in % (Jensen & Allen, 2016). To indicate the dimensional unit of SM in this paper, we will use VWC %. Capacitance sensors measure SM indirectly, in which soil dielectric permittivity measurements are associated with VWC values (Jensen & Allen, 2016).

Soil matric potential (SMP) is a measure of soil water availability to plants. It constitutes the force in which the soil matrix (soil particles and pore space) holds water, i.e., it can be interpreted as the difficulty of a plant to access water in a particular soil. It is measured in *kPa* by a tensiometer (Kashyap & Kumar, 2021). It is also known as soil water tension (SWT) (Rawls et al., 1991; Van Genuchten, 1980).

2.2. Domain-knowledge for soil water behavior

Soil water behavior is composed of the characteristic properties of the soil and the elements that influence soil water dynamics over time. Soil properties data are those related to soil structure (e.g., ground density) and soil water retention curve (SWRC), the latter being a map of hydraulic behavior for a particular soil through water content levels (Rawls et al., 1991; Van Genuchten, 1980). The elements that most influence soil water dynamics are precipitation, irrigation, soil water evaporation, and plant transpiration (Jensen & Allen, 2016).

Soil water retention curve (SWRC) is an empirical curve that relates, for each soil, the SM measured in volumetric water content (VWC) with the soil matric potential (SMP), thus serving as a conversion between these two ways to relate water and soil. Therefore, if you cannot, for economic reasons, sense SM and SMP simultaneously, you can only sense one of the two: with an accurate SWRC, you can convert one into the other. SWRC is usually different for each soil matrix, and there may be relevant variations even within the same field. Its parameters are obtained by in situ or agronomic lab tests (Rawls et al., 1991; Van Genuchten, 1980). In addition, SWRC indicates the points of maximum soil water retention, known as field capacity (FC), and those in which the plant suffers water stress, known as wilting point (WP) and permanent wilting point (PWP) (Rawls et al., 1991; Van Genuchten, 1980).

Water is not a constant in the plant root zone. With time, precipitation and irrigation are primary water sources for the soil, while evapotranspiration is the main responsible for water consumption. Evapotranspiration (ET) combines soil water evaporation (direct effect

from atmospheric conditions) and plant transpiration (side effect from water use by plants photosynthesis). These two factors are considered together for two reasons: they are difficult to be modeled separately, and because both respond together for most soil water outtake. There are several physical-empirical models to it, and they are dependent on weather and crop data or radiation fluxes measurements (Jensen & Allen, 2016).

The most common models calculate a reference ET for a hypothetical standard crop-based uniquely on weather data. Two of them are the FAO-56 Penman-Monteith (FAO [1]standard) (Allen, Pereira, Raes, & Smith, 1998; Jensen & Allen, 2016) and Hargreaves (Jensen & Allen, 2016) methods. Then, this value is updated based on coefficients for actual crop phenology (i.e., growth stage) and plant water stress, resulting in the actual crop ET. Different actual conditions alter the magnitude and the balance between evaporation and plant transpiration. Crop phenology coefficient, or just crop coefficient, is usually agronomic and statistically defined for each crop type considering the time since the crop started (Jensen & Allen, 2016).

LAI (Leaf Area Index) and NDVI (Normalized Difference Vegetation Index) are alternative sources for crop phenology and plant water stress information and can be adapted to generate actual crop ET estimates. LAI is a measurement of plant development (Alvino & Marino, 2017; Carlson & Ripley, 1997; Oca & Flores, 2021; Sishodia, Ray, & Singh, 2020). NDVI is used to analyze natural or agricultural vegetation status (Alvino & Marino, 2017; Carlson & Ripley, 1997; Oca & Flores, 2021; Sishodia et al., 2020). Recently, machine learning approaches have also been applied to crop phenology determination (Pattathal V & Karnieli, 2022).

Both reference and crop ET can be forecasted. As reference ET is weather data-dependent, it can also be forecasted based on the weather forecast (Afzaal, Farooque, Abbas, Acharya, & Esau, 2020; Torres, Walker, & McKee, 2011). And, as crop phenology is a slow-changing factor (Jensen & Allen, 2016), crop ET can also be forecasted based on crop phenology and reference ET (Jensen & Allen, 2016).

2.3. Weather forecast model

Weather forecast is essential to predict future soil behavior better. In this paper, we use two sources of weather forecast: WRF and Agrosmart-proprietary models. The weather research and forecasting (WRF) model is a mesoscale numerical weather prediction system designed for both atmospheric research and operational forecasting applications, maintained by USA NCAR (National Center for Atmospheric Research) (UCAR, 2021).

2.4. Mechanistic approaches to water need estimation (WNE)

The mainstream ways to estimate irrigation water needs are the long-established mechanistic approaches. They simulate future behavior relying on physical-empirical models, i.e., domain-knowledge features. Two of the most used approaches are HYDRUS and AQUACROP.

HYDRUS-1D/2D/3D are finite-element models for simulating the one- and two- or three-dimensional movement of water, heat, and multiple solutes in variably saturated media, respectively. They numerically solve physical models, as the Richards' equation (Farthing & Ogden, 2017) for saturated–unsaturated water flow and convection–dispersion type equations for heat and solute transport (Šimůnek, van Genuchten, & Šejna, 2008, 2016).

AQUACROP is a crop-water productivity model developed by the Land and Water Division of FAO. It calculates evapotranspiration (ET) according to the FAO Penman-Monteith equation. And it uses the growing degree days (GDD) (Miller, Lanier, & Brandt, 2001) as the internal default clock to account for the effects of temperature regimes on crop phenology (FAO, 2021a; Linker et al., 2016).

2.5. Data-driven algorithms

Throughout the work, we use a feature selection method, stepwise forward, and several machine learning (ML) algorithms to support ML-based feature engineering and soil moisture (SM) forecast approach. The standard forward stepwise is a feature selection method that involves starting with no features in the model, testing the addition of each feature using a chosen model fit criterion, adding the feature (if any) whose inclusion results in the most statistically significant improvement of the fit, and repeating this process until none improves the model to a statistically significant extent (Chandrashekar & Sahin, 2014). We can use several metrics to evaluate the model fit for the feature selection task, but here we choose R^2_{adj} (section 2.6) and AIC. Akaike information criterion (AIC) is a metric about the quality of a statistical model, combining error estimation and model simplicity (the smaller the number of features, the better) (Chandrashekar & Sahin, 2014; McElreath, 2020). It, therefore, provides a metric for comparing and selecting models.

For ML to feature engineering, we use autoencoder. It is a type of artificial neural network (ANN) used for unsupervised learning. The autoencoder learns a representation (encoding) for a dataset, typically for dimensionality reduction, by training the network to ignore insignificant data ("noise"). The encoding is validated and refined by attempting to regenerate the input from the encoding (Amarbayasgalan, Pham, Theera-Umpon, & Ryu, 2020; Goodfellow, Bengio, & Courville, 2015). In our case, we used the autoencoder to capture the typical intraday behavior of the monitored variables to calculate later the average error for each day against this typical behavior. Such error measure gives us how much a given day is with an atypical but true behavior, being a way of context-awareness (section 5.3). This feature is important for our ML-based SM forecast, as will be seen in section 6.2.

For ML to SM forecast, we use linear regression (LR) (Freedman, 2009) and decision tree (DT) (Kubat, 2018) to serve as baselines, as they are simple algorithms. We opt to test decision tree ensembles as they are benchmarks for time series problems (Géron, 2019; Gumière et al., 2020; Dubois et al., 2021; Makridakis, Spiliotis, & Assimakopoulos, 2020): random forest (RF) (Géron, 2019) as a representative of independent predictors strategy, and light gradient boosting machine (LightGBM) (Ke, Meng, Finley, Wang, Chen, Ma, & Liu, 2017; Microsoft, 2021a) as a representative of sequential predictors strategy. Additionally, we cover artificial neural networks (ANN) alternatives, as MLP (Géron, 2019; Goodfellow et al., 2015), LSTM (Géron, 2019; Goodfellow et al., 2015), and StemGNN (Cao D., et al., 2020; Microsoft, 2021b). Multilayer perceptron (MLP) is a fundamental approach for neural networks regression and plays an ANN baseline role. Long short-term memory (LSTM) and spectral temporal graph neural network (StemGNN) are two architectures for recurrent neural networks (RNN), an ANN class specialist in time series problems. LSTM is a classical and experienced approach (Goodfellow et al., 2015; Adeyemi et al., 2018; Gumière et al., 2020; Cao D., et al., 2020; Yu et al., 2021; Ahmed, et al., 2021; Wang, et al., 2022), and StemGNN is a promising and recent one, both having a good track record for time series (Cao D., et al., 2020). For all ANN algorithms above, we use rectified linear unit (ReLU) as neuron activation function and stochastic gradient descent (SGD) as optimizer (Goodfellow et al., 2015).

Some related work uses alternative ML algorithms to SM forecast. Support vector machine (SVM) is a classical algorithm that uses a set of hyperplanes of high-dimensional spaces to attend classification or regression problems (Géron, 2019). Adaptive neuro-fuzzy inference systems (ANFIS) is a classical technique that integrates artificial neural networks with fuzzy logic to approximate nonlinear functions (Jang, 1993). A convolutional neural network (CNN) filters data to find

[1] FAO: Food and Agriculture Organization of the United Nations (FAO, 2021b).

patterns and reduce dimensionality, commonly used for image processing and other applications such as feature engineering and dimensionality reduction layers (Géron, 2019; Goodfellow et al., 2015). Gated recurrent unit (GRU) is another recurrent neural network, basically a modified LSTM (Géron, 2019; Goodfellow et al., 2015). CEEMDAN is an acronym for complete ensemble empirical mode decomposition with adaptive noise, a method of time-series spectral decomposition and reconstruction of the original signal (Torres, Colominas, Schlotthauer, & Flandrin, 2011).

2.6. Machine learning metrics

To evaluate SM forecast models, we use key metrics for supervised machine learning (Scikit-Learn, 2021a): RMSE (Scikit-Learn, 2021b), MAE (Scikit-Learn, 2021c), R^2_{adj} (Scikit-Learn, 2021d), and MAPE (Scikit-Learn, 2021e). Root mean square error (RMSE), mean absolute error (MAE) and mean average percentual error (MAPE) have auto-explanatory names (Géron, 2019). R^2 i.e., coefficient of determination is the proportion of the variation in the target variable that is predictable from the features. Adjusted R^2 (R^2_{adj}) is an attempt to account for the phenomenon of the R^2 automatically and spuriously increasing when extra explanatory variables are added to the model (Scikit-Learn, 2021a, 2021d). RMSE and MAE calculate error in our target variable dimension unit, while R^2_{adj} and MAPE are relative metrics.

2.7. Irrigation water depth calculation

The irrigation system should receive water need estimates in water volume. However, at the end of a machine learning-based soil moisture (SM) forecast (section 5.1), the closest we get to this is the measure of SM in volumetric water content, which is a relative measure. To arrive at a measure of absolute volume, we must first calculate the gap between the current and an ideal value of SM weighted according to the soil depth profile for the plant root zone (Toscano, Stanghellini, Bittelli, Castaldi, Soininen, Torre Neto, & Ricchi, 2019). Then, such gap, in relative volume (e.g., VWC in cm^3/cm^3), must be transformed into absolute volume (e.g., cm^3) considering the soil volume to the effective depth of the plant root (Toscano, et al., 2019). Finally, we can transform this value into a usual dimensional unit in agronomy, *mm*, representing the water column height for an area of 1 m^2, which would be equivalent to l/m^2 (Toscano, et al., 2019). Thus, even though water content is measured in soil matric potential, SMP [*kPa*], it is usually convenient to convert it to soil moisture, SM [VWC %], through the soil water retention curve (SWRC) (section 2.2). This is because SM [VWC %] is directly used to calculate irrigation water depth.

3. Related work

In recent years, machine learning-based soil moisture (SM) forecast has become an active research topic in precision irrigation. Karandish and Šimůnek (Karandish & Šimůnek, 2016) propose a comparison between a traditional physical approach, HYDRUS-2D, and ML approaches, such as linear regression (LR), support vector machine (SVM), along with adaptive neuro-fuzzy inference systems (ANFIS). HYDRUS-2D ranks first for multi-depth SM forecast for the following days, but ANFIS and SVM better suits water stress conditions when there is a lack of input data required by HYDRUS-2D. This is a significant result in favor of ML-based approaches, as the considered input data comprises features required by HYDRUS-2D (mainly domain-knowledge features), which represents a natural favoring of the HYDRUS-2D performance. A limitation of that work is that, although the tested models rely on meteorological data such as pan evaporation [2] and average air temperature, they do not consider lagged SM and weather data from sensors. Another limitation is the absence of geographic and crop type diversification, which reduces the generalization of its conclusions: it monitors a single field and a single crop type (maize).

Adeyemi *et al.* (Adeyemi et al., 2018) propose the use of long short-term memory (LSTM) to forecast D + 1 (one day ahead) SM based on past SM, precipitation, and weather data. Using data from three sites, they obtained a R^2_{adj} value of above 0.94 in all sites, a performance level comparable with the other papers referenced in this section. In addition, the resulting models are also able to generate robust SM forecasts for "out-of-sample" sites, which indicates a potential for generalization. Another highlight of this paper is that it uses models to generate irrigation water need calculations through the AQUACROP solution and compares the results with the rule-based irrigation system that applies irrigation based on predefined thresholds. Results indicate a water-saving between 20 and 46% for a similar crop yield, pointing out that ML-based approaches can be much superior in water efficiency, at least against basic irrigation systems as the rule-based. Two limitations of this work are that it relies on data from a single crop type (potato) and that recurrent approaches (as LSTM) are not robust for data with frequent discontinuities (section 6.3) (Goodfellow et al., 2015).

Gumière et al. (2020) compare random forest (RF), with HYDRUS-2D, in forecasting soil matric potential (SMP) in the root zone. After training on a dataset collected in a cranberry field, they evaluated the accuracy of the two models for 30 different time frames within 72 h. The results highlight that both models can accurately forecast the SMP. The ML-based model can achieve better performance when compared to the physics-based model. Still, forecasting accuracy decreases sharply toward the end of the 72 h lead-time, while HYDRUS-2D does not. This is an evidence that ML-based approaches can outperform traditional physics-based benchmarks for the short-term forecast when sourced with SM and weather fine-grained data. The use of SMP as the target variable is valid, but as we will see in section 5.1, it is not the best alternative for obtaining SM forecasts for irrigation water need estimation. Another limitation of this work is that they rely on data from a single site and crop type, making their findings less generalizable across geography and crop types.

Yu et al. (2021) propose a hybrid CNN-GRU for forecasting SM for different depths and time horizons until ten days. They use weather, SM, and soil properties data from five sites of a maize-growing area. The hybrid CNN-GRU outperforms CNN and GRU standalone approaches, achieves an accuracy compatible with the other papers referenced. The model forecasting accuracy improves with soil depth, an expected result as superficial SM is more unstable. The superficial layer is more susceptible to water consumed by evaporation (due to air weather dynamics) and plants, and it also suffers greater impacts because of precipitation (Xu et al., 2017; Zeitoun et al., 2021). Two limitations of this work: it relies on data from a single crop type; and GRU is a recurrent approach that would not be robust when data suffers from frequent discontinuities (section 6.3) (Goodfellow et al., 2015).

Dubois, Teytaud, & Verel (2021) tested different ML approaches (multilayer perceptron – MLP, random forest – RF, and support vector machine – SVM) to forecast soil matric potential (SMP) of a potato crop for up to a week. Alongside SVM, RF outperformed MLP, which corroborates our results, indicating the superiority of decision tree-based ensembles over MLP (section 6.3). They also highlighted the importance of feature selection methods to build simple models, dependent on fewer features and data sources for financial savings and operational

[2] Pan evaporation is an empirical measurement of evaporation rates, through the verification of volume variation in a pan with water exposed to the environment (unless precipitation) (Allen et al., 1998).

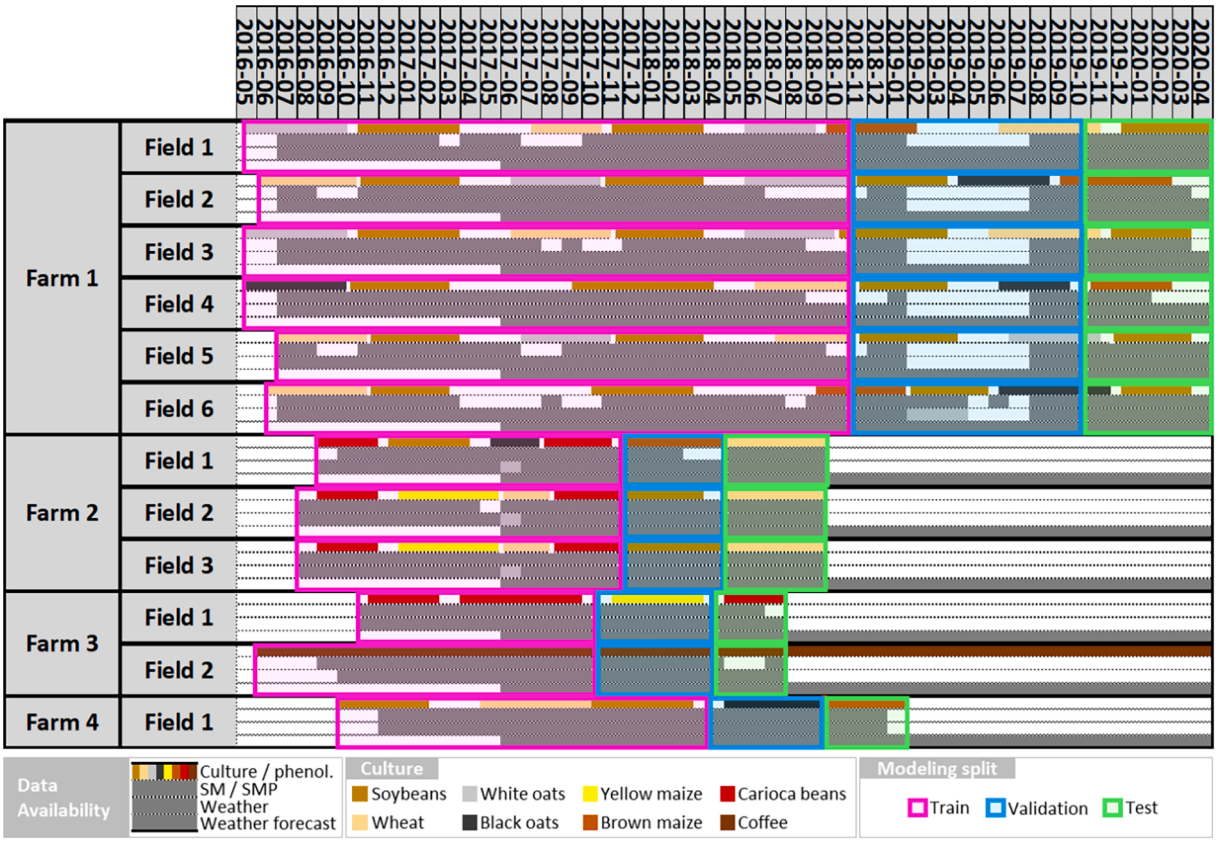

Fig. 1. Data availability landscape and ML train, validation, and test splitting.

ease. As a limitation, this work uses SMP as the target variable, which we showed is not the best alternative (section 6.1). Furthermore, this work relies on data from a single site and a single crop type (potato).

Ahmed et al. (2021) designed a deep learning hybrid model (CEEMDAN-CNN-GRU) for daily SM forecast based mainly on satellite remote sensing. They obtained an accuracy comparable to the other papers from this section, which means remote sensing can also be an important data source. Our paper does not use remote sensing data, but this is an option for soil probe data discontinuities or decalibration (section 6.3). Of course, remote sensing has its issues of coverage area, periodicity, image resolution, and cost, which often do not allow reaching the level of spatial–temporal fine-granularity of in situ IoT solutions (Alvino & Marino, 2017; Beck et al., 2021; Mohanty, Cosh, Lakshmi, & Montzka, 2017). Still, they can certainly act in a complementary way to help resolve IoT data quality issues (Abowarda et al., 2021; Campo, Ledezma, & Corrales, 2020; Jarman & Dimmock, 2018; Sun, Di, Fang, Guo, Tan, Jiang, & Shen, 2021).

Wang et al. (2022) also used CEEMDAN as a signal decomposition technique together with a time series model as LSTM to produce a probabilistic precipitation forecast. Their results are interesting as precipitation is the most challenging factor to predict and the most impactful in the soil moisture. Furthermore, they show the adequacy of RNN models for forecasting time series related to the soil moisture application when there are no data availability problems.

Although each paper above in this section has its application characteristics (input data, locality, and crop type), and, as a result, the performances of their approaches are not perfectly comparable, we will show that our results are in line with the others. That is, we obtained an R^2_{adj} oscillating between 0.94 and 0.98 for D + 1.

Recent work has shown that context-awareness plays a relevant role in SM forecasting. Matei, Rusu, Bozga, and Pop-Sitar (2017) added context data publicly available from web portals to local data, provided by weather stations, to forecast SM, proving that quantity and quality of context data are essencial to the prediction accuracy. Avram, Matei, Pintea, Pop, and Anton (2019) showed that context modeling could add helpful information to SM forecast, improving accuracy by 30%. Avram, Matei, Pintea, and Pop (2020) investigated the impact of context quality (completeness and accuracy) over predictive forecasting for soil moisture in context-aware data mining. They concluded that the SM forecast improves when using context-awareness over not using it, even if the quality standards are not completely met.

In summary, compared to related work, our paper brings three main innovative aspects:

a) We propose ML good practices for data-driven SM forecast and organize modeling planning steps in a reference guide for solution designers. In this sense, we address sensitive topics such as: whether it would be better to use volumetric water content (VWC) or SMP as target variable; what are the critical feature groups (and data sources, as a consequence); how weather and precipitation forecast help to improve accuracy; how sensing context-awareness can be modeled and act as an essential predictor; what are the suitable ML algorithms and under what conditions should they be used; how ensembles can be modeled to improve accuracy; how weighted training can influence accuracy on the hardest-to-forecast days; how weighted metrics can be helpful to interpret performance against uncertainty factors (as future weather and precipitation).
b) We provide an alternative to solely using domain-knowledge features (evapotranspiration, crop phenology, and soil properties models) as input data with an entirely data-driven approach. This approach is made possible by a context-awareness feature that implicitly captures domain-knowledge information.
c) We examine the impact of IoT data quality issues on ML-based SM forecasting and explore alternatives to deal with these issues.

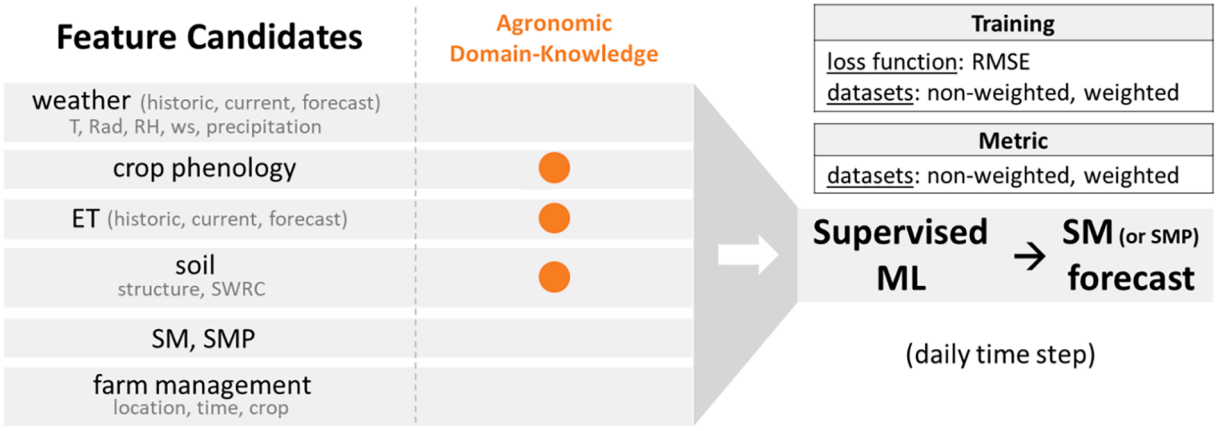

Fig. 2. Our ML-based SM forecast general approach.

Our results are supported by data over two to four years from twelve fields from four geographically and climatically diverse farms in Brazil, working with 55 crops of eight different crop types. Such diversification allows for a certain generalization of our findings.

4. Data description

Data for this research is provided by Agrosmart[3], a Brazilian smart irrigation supplier currently present in 25 Brazilian states, monitoring around 550,000 ha of 55 different crops. Data comes from four different farms. Farm 1, with six fields, and farm 4, with a single field, are in south Brazil, where the prevailing climate is humid subtropical. Farm 2, with three fields, is in southeast Brazil, where the prevailing climate is tropical. And farm 3, with two fields, is located near midwest Brazil, where the prevailing climate is tropical savanna. In all, the data covers eight different crop types: soybeans (26 crops), wheat (15 crops), white oats (8 crops), black oats (6 crops), yellow maize (3 crops), brown maize (6 crops), pinto beans (9 crops), and coffee (single-field permanent crop). Such climate and crop type diversification, combined with soil moisture (SM) forecast accuracy independent of farm-field location and crop type (section 6.2), makes our findings generalizable across the covered geographies and crop types.

The evaluated areas have climate Köppen classification of Aw and Cfa (Dubreil, Fante, Planchon, & Neto, 2018; Martins, Gonzaga, Santos, & Reboita, 2018; Wrege, Steinmetz, Júnior, & Almeida, 2012). Considering that the area of the world suitable for agriculture is composed of the biomes tropical rain forest (TRF), grassland, subtropical scrub and woodland (SSW), subtropical and mid-latitude forest (SMF), and tundra biomes (Cui, Liang, & Wang, 2021) and the percentage of Aw and Cfa climates in each of these biomes (Rohli, Joyner, Reynolds, & Ballinger, 2015), we have that 25.5% of the worlds' area suitable for agriculture has a climate similar to the areas evaluated.

Fig. 1 shows the arrangement of farms, fields, and crops in time. For each field, the top row contains the crops, where the horizontal bar colors represent crop types. The second to fourth rows indicate where soil moisture (SM), weather, and weather forecast data are available. Our SM measures are converted from the soil matric potential (SMP), collected by tensiometers in the soil probes. Data covering 2–4 years of farm activity allows exposure to annual seasonality. Depending on the field and the time, the original sensor data (SMP and weather) typically has 1 to 10 measurements per hour. SMP data is captured in 3 depths (20*cm*, 40*cm*, and 60*cm*), hence soil water profile is modeled as a feature. Weather data come from in-farm weather stations, and weather forecasts come from an ensemble between water research and forecasting (WRF) model and Agrosmart-proprietary models.

As shown in Fig. 1, sensing data is subject to frequent discontinuities, mainly from soil probes data, due to equipment or field communication failures. Unavailability of sensing data is an issue, as SM forecast is a data-driven approach. The lack of SM data is particularly detrimental since it provides the most significant predictive power on future SM (section 6.2). Furthermore, relevant machine learning (ML) time-series techniques, such as recurrent neural networks (e.g., LSTM and StemGNN, both tested in this research), have their performance impaired by discontinuities (section 6.3) because their mechanics assume temporal continuity (Goodfellow et al., 2015). Thus, one of our recommendations for ML-based SM forecast is to pay special attention to sensing data quality, whether through robust IoT solutions or redundancy in data sourcing (Togneri, Camponogara, & Kamienski, 2019). The first four columns of **Table A-1** in the Appendix provide all the data details: feature class, feature, description, and values dominium (dimensional unit for quantitative features, and categories for qualitative ones).

5. Methodology

5.1. General approach for soil moisture (SM) forecast

5.1.1. Modeling data and flux

Fig. 2 illustrates our SM forecast ML (machine learning) approach. For model input, we consider both raw data and domain-knowledge-engineered features. Raw data are those directly collected from farm management entries or sensors. Farm management data comprises location (farm-field, latitude, and longitude), timestamp (time of the year, to capture seasonality), and crop records (crop type, start date, and end date). Sensor data includes SM, soil matric potential (SMP), and weather (air temperature (T), solar radiation (Rad), relative humidity (RH), wind speed (ws), and precipitation). Domain-knowledge-engineered features come from agronomic knowledge (physical and empirical models), such as crop phenology, evapotranspiration (ET), and soil properties. Our crop phenology data are agronomic and statistically defined coefficients for each crop type considering the time since the crop start date (Jensen & Allen, 2016). We do not have common remote sensing data as LAI and NDVI, which are usually used to infer crop phenology. That is why we used traditional statistically defined coefficients. If we had LAI or NDVI, crop phenology could also be considered raw data input as LAI and NDVI are standard collected data. Reference evapotranspiration is calculated from weather data by the FAO-56 Penman-Monteith (FAO standard) and Hargreaves methods. Also, crop evapotranspiration is derived from reference evapotranspiration and crop phenology coefficients. Soil properties data are related

[3] Data used in this paper may be avaible upon request, under a non-disclosure agreement.

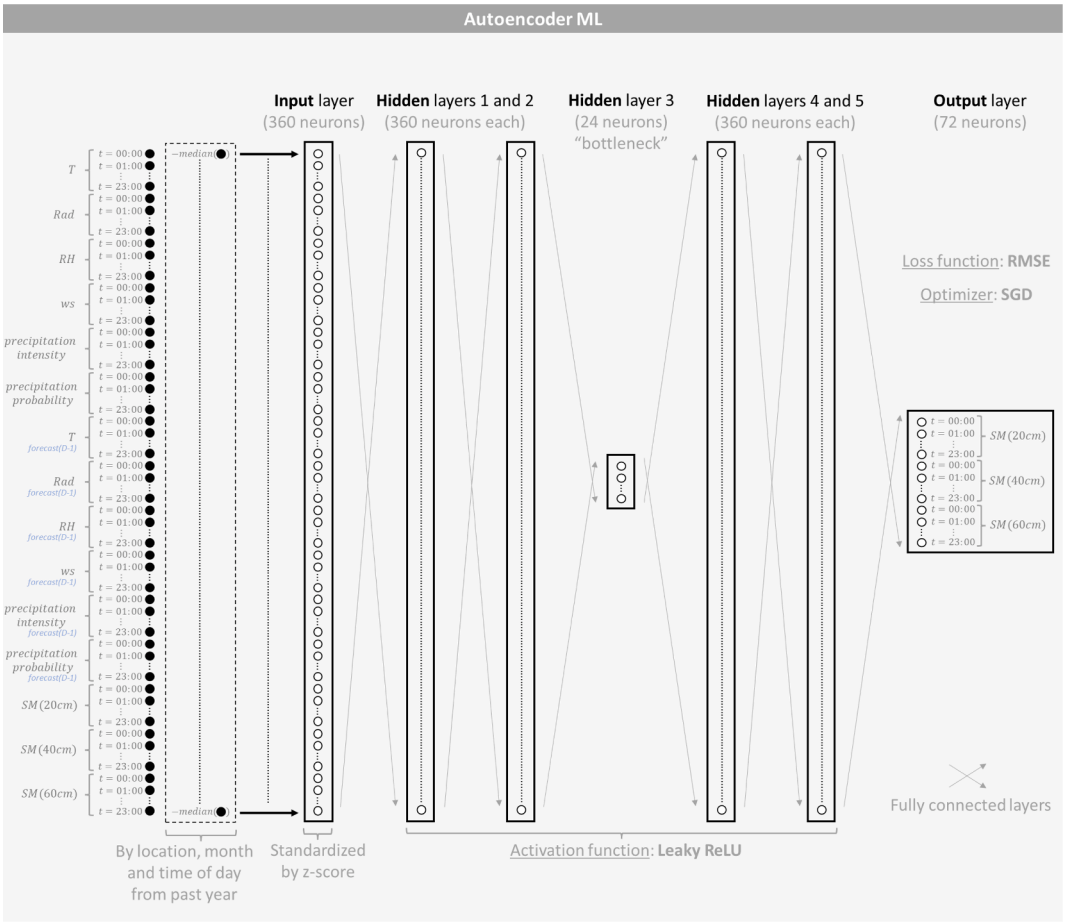

Fig. 3. Autoencoder architecture.

to soil structure (e.g., ground density) and soil water retention curve (SWRC) with its parameters. All features have their scales standardized by z-score, except for decision tree-based algorithms because they do not require scale standardization (Géron, 2019). Time features are transformed via cyclic components such as sine and cosine to help ML capture seasonal behavior better (Géron, 2019).

We use supervised ML algorithms (see sections 5.6 and 5.7) to provide SM forecast or SMP forecast daily. It is possible to run the SMP forecast and then convert their values to SM through the soil water retention curve (SWRC), so that the SM forecast is obtained at the end (section 5.4). A daily time step is chosen for three reasons: it is the time step used by most related work in SM forecast (section 2.1: (Adeyemi et al., 2018; Yu et al., 2021; Dubois, Teytaud, & Verel, 2021; Ahmed, et al., 2021)); because irrigation modalities for large-scale crops imply periodicity restrictions of irrigation opportunities (e.g., central pivot passes through the same area every 12–24 h typically, depending on the speed imposed) (Reddy, 2017); and because it is a common practice in many countries (including Brazil) for irrigation to be carried out once a day at night, when lower electricity rates are applied (Abrishambaf et al., 2020; García et al., 2016). Thus, the original sensor data (SM and weather) were aggregated by day, using statistics such as minimum, maximum, open (first value of the day), close (last value of the day), mean, and standard deviation. For more details on which statistic or measurement we use as feature candidates, see **Table A-1** (Feature column) in the Appendix. For all models developed in this paper, we use RMSE as the ML loss function and eventually use MAE, R^2_{adj}, and MAPE as evaluation metrics for purposes other than model training. Hyperparameter tuning is performed using Bayesian optimization (Pelikan, Goldberg, & Cantú-Paz, 1999), implemented using a Gaussian process in gp_minimize function (n_initial_points = 20, n_calls = 50) of scikit-optimize library (skopt) (scikit-optimize, 2022). **Table A-2**, in the Appendix, provides the hyperparameter space for all ML algorithms used in this paper.

5.1.2. *Dataset non-weighted and weighted setups*

ML models are trained and have their performance measured in both non-weighted and weighted setups, as Fig. 2 shows. The non-weighted setup serves as a baseline comparison with the models of the papers cited in section 2.1. In this setup, we train and evaluate performance without any weights applied to data points. On the other hand, the weighted setup increases the representativeness of critical data points to 50% of the performance measures, i.e., critical data points receive weight such that they represent 50% of the metric. Critical data points are the days when precipitation occurs or when the weather forecast indicates the possibility of precipitation. This is because precipitation is the event that disrupts the SM, and uncertainty about precipitation makes prediction difficult (Cai, Hejazi, & Wang, 2011; Cao, Tan, Cui, & Luo, 2019; Shang, Chen, Stroock, & You, 2020). As for non-critical data points, SM forecast is easier because it primarily depends on evapotranspiration (ET), whose behavior is smoother in time (Jensen & Allen, 2016). Thus, in a weighted setup, we allow closer attention to critical data points in training. Besides, we also verify that attaining performance on the critical data points is more challenging when compared with overall performance under a non-weighted setup (sections 6.1 to 6.4).

Weighting can be used independently in training the model and measuring the performance in the validation and test datasets. For example, we can train the model with the weighted setup and measure the final result in both weighted and non-weighted setups. The same is

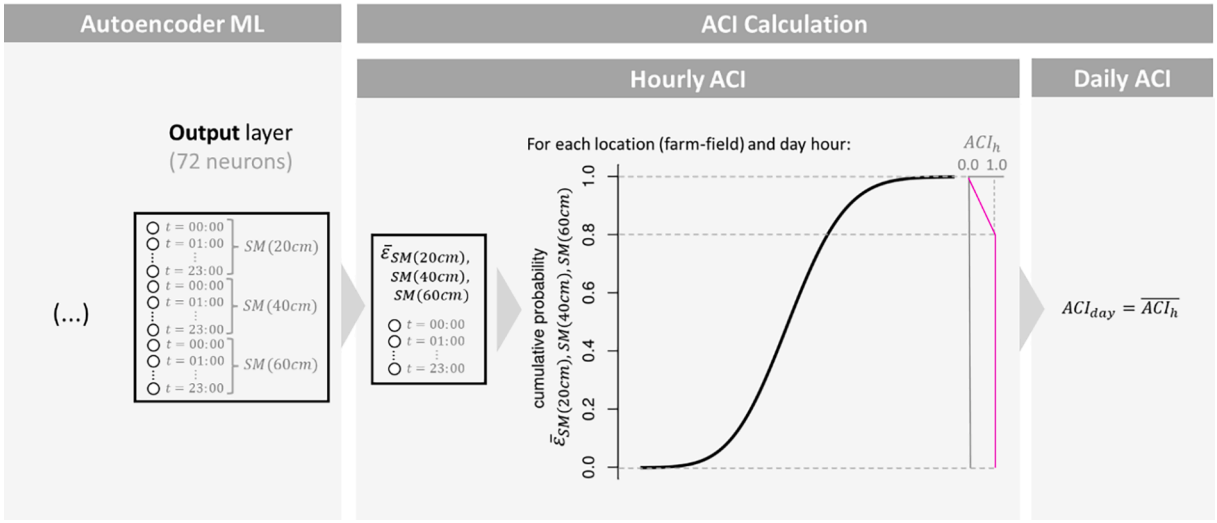

Fig. 4. Abnormal Context Index (ACI) calculation.

valid for the non-weighted setup trained model.

5.2. Train, validation, and test splitting

Fig. 1 shows how we split the dataset into training, validation, and testing. To keep temporal ordering within each field (since it is a time-series forecasting problem), and, as much as possible, have each of the crop types represented in each subset, we choose standard holdout as the validation strategy (Roberts et al., 2017), assuring that the validation and test data are at the end of the series. If we decided on a strategy that further partitioned the training dataset, such as time-series *k*-fold (or any other applicable cross-validation) (Roberts et al., 2017), we would be privileging some crop types over others in each training process, which would make it harder for the model to learn crop type traits. By performing the standard holdout, we can obtain a training dataset that considers all crop types, and, at the same time, we can ensure that almost all crop types are represented in the validation and testing datasets, even in different farms and fields from those observed in the training dataset. The fact that some crop types are represented in different farms and fields among training, validation, and testing datasets, added to the fact that our models obtained acceptable performance compared to the benchmarks (section 6.4), and that the models are crop type independent (section 6.2) reinforces that our models generalize well to different crop types. We can also justify the non-use of cross-validation a posteriori through the verified stability of accuracy between validation and testing (section 6.3).

The cutoff dates between the training, validation and testing periods were the same for all fields within each farm to avoid data leakage. Each farm possesses an independent weather behavior, as they are geographically distant from each other and represent different climatic regions (section 4). The fact that, for example, the same date is in a training dataset for a farm and validation for another does not represent data leakage. This division was also necessary to obtain a significant volume of data for training, validation, and testing in all farms since the data from some farms (farms 2, 3, and 4) cover a considerably shorter period than that of the other (farm 1).

Within training, validation, and testing intervals, the data points effectively considered were those in which the data is complete, i.e., no missing values in SM and weather data. When a model used weather forecast variables, data points without them were also left out (weather forecast data only have history from mid-2017).

5.3. Context-awareness

Context-aware features are engineered to improve the forecast for critical data points. We use two coupled components: an asymmetric autoencoder, and the subsequent calculation of what we call the Abnormal Context Index (ACI).

The autoencoder captures the typical intraday behavior of soil moisture to calculate the average error for each day against this typical behavior. Fig. 3 illustrates its architecture. It is considered asymmetric because the input layer is different in size from the output layer. In the output layer, as target variables, we only have soil moisture data, while in the input layer, we also have meteorological data. To be more precise, the input layer considers: (i) weather (air temperature (T), solar radiation (Rad), relative humidity (RH), wind speed (ws), and precipitation); (ii) weather forecast (same features as for weather: T, Rad, RH, ws, precipitation intensity, precipitation probability); and (iii) soil moisture (at 20 cm, 40 cm, and 60 cm depth) for each hour in a day lined up as a 1D-array. This strategy makes an input layer of 360 neurons. In the output layer, as only soil moisture data (at 20 cm, 40 cm, and 60 cm depth) are considered, and as there are 24 datapoints for each soil moisture data representing each of the day hours, the layer has 72 neurons as a result. Autoencoder input is not the original data, but the difference between it and the median behavior of previous years for each location, time of year (month), and time of day, to deduct systematic effects. We tested several architectures, with the best results attained with five dense intermediate layers, all with the same input vector length (360 neurons), except for the central layer, which served as a bottleneck with 24 neurons. All features are standardized by z-score, the activation function is leaky ReLU, the loss function is RMSE, and the optimizer is SGD.

From the residuals of the autoencoder, we calculate the ACI, which is a heuristic standardization of the residuals. Fig. 4 illustrates the calculation process. We consider the cumulative distribution of residuals up to the 80% level to be normal and assign a value of 1. From 80% to 100%, we assign values ranging from 1 to 0 for ACI, linearly distributed. The ACI goal is to measure how much the hourly intraday behavior differs from the expected behavior, as the median for that location, time of year and time of day. A low ACI, close to 0, indicates a rare multivariate behavior in the form of context-awareness, which, in turn, can be important information for the SM forecast model to perform better. A high ACI, close to 1, indicates typical behavior. ACI is calculated as autoencoder residuals aggregation, segmented both: (i) hourly, to monitor when possible anomalous behaviors occur throughout the day; and (ii) daily, for more general monitoring. The ACI measure gives us

Table 1
Target variable definition summary.

Dimensions of target variable definition	Literature mainstream	Our choice	Type of choice
Time step	daily	daily	domain-dependent
Time horizon	D + 1, D + 2	D + 1, D + 2	domain-dependent
Statistics	Single point (CLOSE, MEAN[13])	MIN, MAX	domain-dependent
Concept	SM [VWC %] SMP [kPa]	Options: SM [VWC %] SM [VWC %] norm. SMP [kPa]	tested

[13] When modeling the mean, the literature usually models or measures the associated variance as well.

Table 2
SM forecast ML modeling scenarios for the study of target variable concept.

Algorithm	Target	Training	Metric
RF	(SM MIN, SM MAX) (D + 1)	loss function: RMSE [SM [VWC %]] (non-weighted dataset) (weighted dataset)	RMSE [SM[VWC %]] (non-weighted dataset) (weighted dataset)

Table 3
Feature groups - Options to SM forecast modeling.

	Historic		Future (forecast: D + 1, D + 2)
	Lagged 5 days	Current (D0)	
weather (except precipitation)	X	X	X
crop phenology and ET	X	X	X
water intake (precipitation + irrigation)	X	X	X
SM, SMP	X	X	Target
ACI	X	X	
Soil		X	
Location		X	
Time		X	
crop type		X	

Feature groups (blocks)
Target: SM (MIN, MAX)

how much a given day is with an atypical but authentic behavior, being a way of context-awareness. This feature is important for our ML-based SM forecast, as presented in section 6.2.

5.4. Target variable definition

The definition of the target variable goes through four dimensions: time step, time horizon, statistics, and concept (Kashyap & Kumar, 2021; Topp, Parkin, & Ferré, 2008; Van Genuchten, 1980). For the first three dimensions, we make our decisions based on application objectives and utilization scenarios. For the last dimension, we take our decision by performing a test. Table 1 summarizes our resultants, comparing our approach with the literature. The reasons for such resultants and the details of the test are set out below.

For the time step, we opt for daily periodicity, as is common in recent literature on soil moisture (SM) and soil matric potential (SMP) forecast (section 2.1) (Adeyemi et al., 2018; Ahmed, et al., 2021; Dubois et al., 2021; Yu et al., 2021). Our decision is justified by the operational issues posed by irrigation equipment for large-scale crops (Reddy, 2017) and the electricity tariff policies (Abrishambaf et al., 2020; García et al., 2016) discussed in section 5.1. These issues independently lead to daily irrigation opportunities.

For the time horizon, we opt for D + 1 (the day after the current day, D0) and D + 2 (two days ahead D0). While this choice meets our application goals, we observe that our dataset contains weather forecasts (including precipitation) of only two days, and those features make up significant factors for accuracy (section 6.2).

For the statistics, we opt to seek minimum and maximum SM values. The literature frequently opts for forecasting a single point by time horizon, being it a measure of central tendency (e.g., the mean of the day) or a value at an arbitrated condition (e.g., the day closing value – i.e., the last value of the day) (Adeyemi et al., 2018; Ahmed, et al., 2021; Dubois et al., 2021; Gumière et al., 2020; Karandish & Šimůnek, 2016; Yu et al., 2021). This practice ignores the SM intraday variability, reaching minimum (MIN) or maximum (MAX) values harmful to the plant. We understand that plant risks occur on both daily extremes: the minimum value indicates the chance that SM turns out below the wilting point (WP) or near to the permanent wilting point (PWP); while the maximum value gives us SM risk of reaching or surpassing field capacity (FC), causing leaching (Jensen & Allen, 2016; Rawls et al., 1991; Van Genuchten, 1980). The daily variance could be associated with daily mean values to estimate probability distribution (minimum and maximum values in consequence). However we understand that it is not better than modeling minimum and maximum value directly, avoiding probability asymmetry modeling complexities.

For the variable concept, we have diverse concepts for measuring water in the soil. The two main ones are SM, usually measured in relative volumetric water content (VWC %), and SMP, measured in kPa (section 2.1). Besides being able to be converted between them by the soil water retention curve (SWRC) (section 2.1), any of them can be used as the target variable of the modeling process (Ahmed, et al., 2021; Dubois et al., 2021). They have different characteristics, as the SWRC establishes a sharp nonlinear relationship between them (Rawls et al., 1991; Van Genuchten, 1980). However, the forecast performance must always be measured using SM [VWC %], the concept and dimensional unit usually used to calculate irrigation water depth (section 2.7). Thus, when used as a target variable, we can test which concept produces the best performance ultimately measured in SM [VWC %]. Following this line, we decided to test as target variable both SM [VWC %] and SMP [kPa], plus a normalized version of SM [VWC %]. The rationale of SM [VWC %] vs. a normalized version of SM [VWC %] follows: as different soils have different WP, PWP, and FC levels, their SM oscillates in different ranges of VWC, allowing the hypothesis that it could make it harder to build a single model to forecast SM for plenty of farms and fields (Rawls et al., 1991; Van Genuchten, 1980). So, we decided to test a normalized VWC, in which the PWP leads to 0% and the FC leads to 100% for each field, independent of the original values of PMP and FC.

Table 2 summarizes modeling scenarios for the study of the target variable concept. To test the best concept, we run an SM forecast model through a random forest (RF) algorithm (Géron, 2019), with D + 1 minimum and maximum values as targets. We chose RF as the machine learning algorithm because it is an effective algorithm for time-series problems (Géron, 2019; Gumière et al., 2020). For all the three target variable concepts, the chosen metric is RMSE of SM [VWC %], to have a comparative basis, and because SM [VWC %] is the variable concept used for later calculations of the irrigation water need (in water volume) (Toscano, et al., 2019). For each target variable, the test is performed considering two dataset scenarios regarding weighting: non-weighted training leading to non-weighted metric and weighted training leading to weighted metric. We run independent models for each target variable and dataset scenario.

Table 4
SM forecast ML modeling scenarios for the feature group selection.

Feature Selection	Algorithm	Target	Training	Metric
Method: Forward Stepwise (adapt. for feature groups) Criteria: AIC	RF	(SM MIN, SM MAX) (D + 1)	loss function: RMSE [SM[VWC %]] (non-weighted) (weighted)	R^2_{adj} (non-weighted) (weighted)

Table 5
SM forecast ML modeling scenarios for algorithm comparison.

Algorithm	Target	Training	Metric
(Naïve, RL, DT, RF, LightGBM, MLP, LSTM, StemGNN)	(SM MIN, SM MAX) (D + 1, D + 2)	loss function: RMSE [SM[VWC %]] (non-weighted, weighted)	(RMSE, MAE, R^2_{adj}, MAPE) (non-weighted, weighted)

5.5. Feature group selection

Feature group selection aims to identify the essential feature groups to soil moisture (SM) forecast so that a solution designer can focus their data sourcing efforts on what is most effective. We adapted the forward stepwise method (Chandrashekar & Sahin, 2014) for our purpose. We have adapted the method so that instead of testing and adding one feature at a time, it tests and includes an entire feature group. Objectively, our goal is to find out the most effective feature groups and data sources.

Table 3 shows these feature groups, represented with 'X' written in the cells. The cell with 'Target' written in it represents the group of target variables. The rows represent different feature classes, frequently associated with diverse data sources. Note that each feature class is further divided by time reference, producing multiple groups. Weather, crop phenology, evapotranspiration (ET), and water intake have two historic blocks, one for the current value (measured in D0 at the end of the day) and another for the lagged five days[4]. In addition, they have future blocks with D + 1 and D + 2 values relying on weather and precipitation forecast. Soil water metrics group, formed by SM and soil matric potential (SMP), also has both historical groups (D0 and lagged five days), but its future group is not eligible because it holds our target variables as we want to perform SM forecast. Abnormal context index (ACI) has both historical groups. Soil, location, and culture contain immutable or slow-changing features such that only the current (last available) values suffice. We also consider just the current value as a reference for the models to perceive eventual systematic seasonal behavior for the time group. **Table A-1**, in the Appendix, expands Table 3, providing more detail on what are the features within each feature group.

Table 4 summarizes modeling scenarios for the feature group selection. We run our adapted forward stepwise method to our SM forecast process, using Random Forest (RF) as a machine learning (ML) algorithm. We again chose RF as the machine learning algorithm as it is considered an effective algorithm for time-series problems (Géron, 2019; Gumière et al., 2020) and because decision trees are themselves natural feature selection methods. Both minimum (MIN) and maximum (MAX) values at D + 1 are considered target variable scenarios. For each of them, the method is performed considering two dataset scenarios regarding weighting: non-weighted training leading to non-weighted metric and weighted training leading to weighted metric. We run independent models for each target variable and dataset scenario. The target value concept is fixed at SM [VWC %], as it is the chosen one after the target variable definition process (section 6.1). We use AIC (Akaike information criterion) as the forward stepwise criteria to determine feature groups and increase the model accuracy while controlling its complexity. We keep RMSE as the RF loss function but choose R^2_{adj} as the metric to evaluate each forward stepwise iteration improvement in the percentage of explainable variance.

The reasons why we chose stepwise forward as the feature selection method are: i) it is relatively lightweight, as it starts with little input data and increases, allowing faster iterations until the stopping criterion is met; this makes a significant difference in processing time, since at each stepwise iteration we perform hyperparameters tuning of the ML algorithm used; ii) we use stepwise forward in partnership with RF, that, as it is based on decision trees, it is an algorithm that naturally already makes feature selection by entropy criteria (Géron, 2019); thus, actually, stepwise forward is not alone: we are using two feature selection methods in a complementary way that increases robustness: stepwise forward and RF.

5.6. Algorithm comparison

Algorithm comparison aims to indicate which machine learning (ML) algorithms are most suitable for our real case and to gather insights about ML algorithms' generalization strengths. We evaluated the following algorithms: LR (linear regression), DT (decision tree), RF (random forest), LightGBM (light gradient boosting machine), MLP (multilayer perceptron), LSTM (long short-term memory)) and StemGNN (spectral temporal graph neural network). As the main baseline, we use a naïve approach, which uses the last available SM (at D0) as predicted values. LR and DT are simple algorithms and serve as an extra baseline regarding the model's complexity dimension: we aim to understand how much accuracy simpler models can provide compared to complex models. Then comes the classes of algorithms we are most interested in, as they are considered state-of-the-art for time-series problems (Borisov, et al., 2022; Cao et al., 2020; Géron, 2019; Gumière et al., 2020): decision tree ensembles and recurrent neural networks (RNN). For decision tree ensembles, we choose RF as the representative of independent and LightGBM as the representative of sequential predictors type. We choose LSTM and StemGNN as representatives of the RNN approaches, which yielded competitive results in time-series problems (Adeyemi et al., 2018; Ahmed, et al., 2021; Cao et al., 2020; Goodfellow et al., 2015; Gumière et al., 2020; Wang, et al., 2022; Yu et al., 2021). We chose StemGNN also because it outperformed N-BEATS [5] in the M4 competition datasets (Cao et al., 2020). M4 is a time series forecasting competition that evaluates algorithms through 100,000 time series (Makridakis et al., 2020). N-BEATS is an algorithm that has previously proven to outperform M4 winners in M4 datasets (Oreshkin, Carpov, Chapados, & Bengio, 2019). Finally, we added MLP as another baseline, a standard approach of artificial neural networks (ANN), without the recurrency component.

Table 5 summarizes modeling scenarios for the algorithm comparison. The features available for the ML algorithms are those of the relevant feature groups, selected according to sections 5.5 and 6.2. The minimum (MIN) and maximum (MAX) values at D + 1 and D + 2 are considered target variable scenarios. For each of them, the method is performed considering two dataset setups regarding weighting: non-

[4] The time window size of 5 days was arbitrarily defined, based on authors' practical experience. Future work can explore other time windows sizes, and even different sizes for each feature.

[5] N-BEATS: Acronym for neural basis expansion analysis for interpretable time series, a time series forecasting algorithm (Oreshkin, Carpov, Chapados, & Bengio, 2019).

Table 6
SM forecast ML modeling scenarios for the algorithm ensemble.

Algorithm	Target	Training	Metrics
Light GBM	(SM MIN, SM MAX) (D + 1, D + 2)	loss function: RMSE [SM [VWC]] (non-weighted, weighted)	(RMSE, MAE, R^2_{adj}, MAPE) (non-weighted, weighted)

weighted and weighted training. At this time, for both non-weighted and weighted trained models, the accuracy is measured in non-weighted and weighted schemes. The goal is to assess whether non-weighted trained models perform properly on the weighted setup (which better represents critical data points) and whether weighted trained models also perform properly on the non-weighted setup (which better represents typical data points). We run independent models for each target variable and dataset scenario. The target value concept is fixed at SM [VWC %], as it is the chosen one after the target variable definition process (section 6.1). We keep RMSE as the ML loss function, but four different accuracy metrics are collected: RMSE, MAE (mean absolute error), R^2_{adj}, and MAPE (mean absolute percentage error). The tuned hyperparameters of the trained models are found in table **Table A-3** in Appendix.

5.7. Algorithm ensemble

We ensemble all the resulting models from section 5.6 in a single predictor for each scenario to capture the best characteristics and, consequently, to pursue a performance improvement (Géron, 2019; Goodfellow et al., 2015). Table 6 summarizes modeling scenarios for the algorithm ensemble. The same algorithm comparison (section 5.6) scenarios are kept for target variable, training, and metric. For each of these scenarios, the predicted values of all tested algorithms, from both non-weighted and weighted training, are considered feature options. We run independent models for each target variable and dataset scenario. The machine learning (ML) algorithm used for the ensemble is LightGBM, as it proves to be the most adequate (better and more stable accuracy) for our real case (section 6.3). The tuned hyperparameters of the resulting ensemble model are found in **Table A-3** in Appendix.

6. Results and discussion

6.1. Target variable definition

Our target variable definition has four decision dimensions. Time-step, time horizon, and statistics are arbitrarily defined, with the respective decisions and justifications exposed in section 5.4. We decide the variable concept by a test, whose methodology is also disclosed in section 5.4 but whose results and discussion are in the present section. The options for target variables are soil moisture (SM [VWC %]), normalized soil moisture (SM [VWC %] norm.), and soil matric potential (SMP [kPa]).

Fig. 5 compares the target variable possible concepts for all proposed scenarios. Even though we are comparing three different concepts of the target variable, the performance metric, RMSE, is calculated in SM [VWC %] for all of them. This is due to two main reasons: first, to have a comparative base among target variable concepts; second, this variable concept is used for later calculations of the irrigation water need in water volume (section 2.7). RMSE is the mean of all errors in D + 1 over all data points of the sliding window on the validation and test datasets. We can see that SM [VWC %] and normalized SM [VWC %] provide significantly lower and more stable RMSE between validation and test datasets in relation to SMP [kPa]. We believe this is because SMP [kPa] vs. SM [VWC %] presents a strong nonlinear relationship (Rawls et al., 1991; Van Genuchten, 1980), which may even be different for each field, so ML empirically can handle SM [VWC %] better than SMP [kPa] as target variable concept. So, we conclude that even though soil moisture (SM) data is collected in SMP [kPa] by soil probes equipped with tensiometers, it pays off to convert the concept to SM [VWC %] to serve as

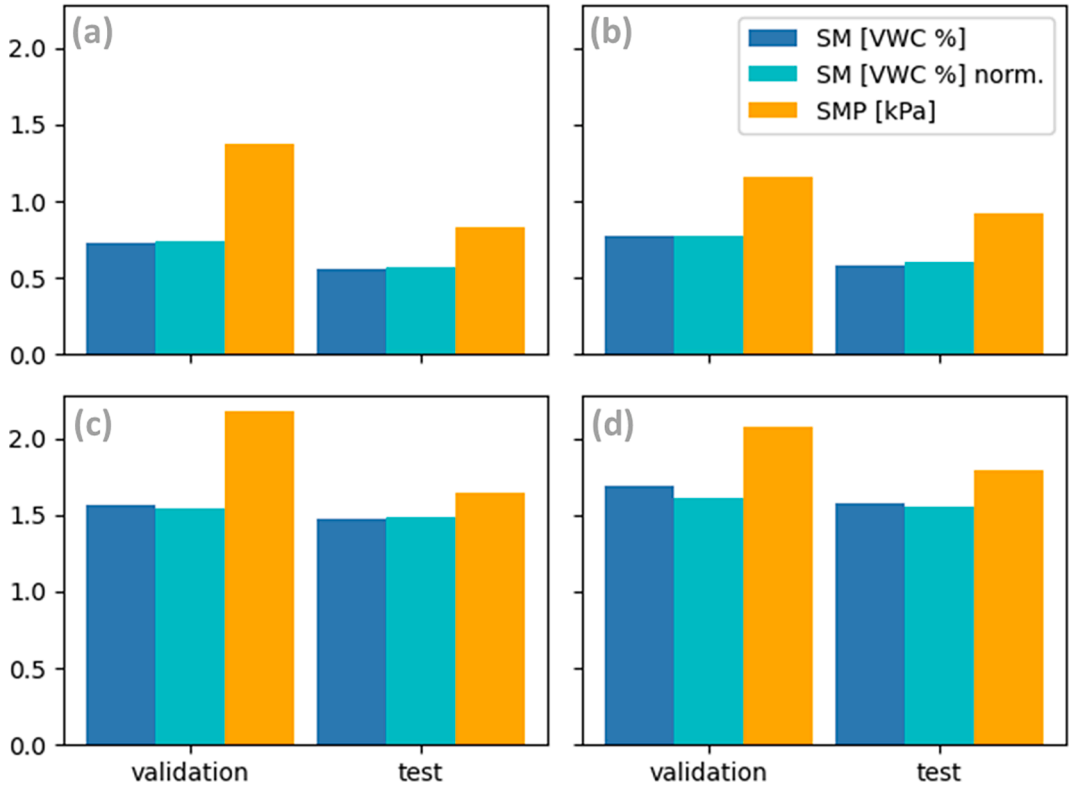

Fig. 5. Target variable definition – Correspondent RMSE [SM[VWC %]] for D + 1 forecast. (a) Target: SM MIN, weighting: NO; (b) Target: SM MIN, weighting: YES; (c) Target: SM MAX, weighting: NO; (d) Target: SM MAX, weighting: YES.

Table 7
Feature selection results by stepwise forward method - Results. (a) Target: MIN, D + 1; Training and metric weighting: NO; (b) Target: MIN, D + 1; Training and metric weighting: YES; (c) Target: MAX, D + 1; Training and metric weighting: NO; (d) Target: MAX, D + 1; Training and metric weighting: YES.

(a)	Validation		Test		(b)	Validation		Test	
Added feature group	R^2_{adj}	$\Delta(R^2_{adj})$	R^2_{adj}	$\Delta(R^2_{adj})$	Added feature group	R^2_{adj}	$\Delta(R^2_{adj})$	R^2_{adj}	$\Delta(R^2_{adj})$
none	0.00	0.00	0.00	0.00	none	0.00	0.00	0.00	0.00
SM, SMP (current) *	0.96	0.96	0.98	0.98	**SM, SMP (current)** *	0.97	0.97	0.94	0.94
SM, SMP (lagged 5 days) **	0.96	0.00	0.98	0.00	**SM, SMP (lagged 5 days)** **	0.97	0.00	0.94	0.00
ACI (lagged 5 days) **	0.96	0.00	0.98	0.00	**precipitation (lag. 5 days)** **	0.97	0.00	0.94	0.00
farm / field **	0.96	0.00	0.98	0.00	**ACI (current)** **	0.97	0.00	0.94	0.00
					farm / field **	0.97	0.00	0.94	0.00

(c)	Validation		Test		(d)	Validation		Test	
Added feature group	$\mathbf{R^2_{adj}}$	$\Delta(\mathbf{R^2_{adj}})$	$\mathbf{R^2_{adj}}$	$\Delta(\mathbf{R^2_{adj}})$	**Added feature group**	$\mathbf{R^2_{adj}}$	$\Delta(\mathbf{R^2_{adj}})$	$\mathbf{R^2_{adj}}$	$\Delta(\mathbf{R^2_{adj}})$
none	0.00	0.00	0.00	0.00	none	0.00	0.00	0.00	0.00
SM, SMP (current) *	0.85	0.85	0.83	0.83	**SM, SMP (current)** *	0.56	0.56	0.59	0.59
ACI (current) *	0.88	0.03	0.86	0.03	**ACI (current)** *	0.69	0.12	0.72	0.13
					precipitation forecast *	0.70	0.02	0.74	0.02
					crop type **	0.70	0.00	0.74	0.00

* High relevance ($\Delta(R^2_{adj}) \geq 0.005$).

** Low but positive relevance ($0.000 \geq \Delta(R^2_{adj}) > 0.005$).

the machine learning (ML) target variable. Thus, we have already ruled out SMP [kPa] as suitable for the target variable concept.

As for the choice between SM [VWC %] and normalized SM [VWC %], the comparison of results was tighter. We can see in Fig. 5 that both had similar RMSE, both in terms of level and stability between validation and test datasets. However, as (i) the variable concept used for irrigation water need calculation is SM [VWC %], and (ii) the normalized SM [VWC %] was created just to check if it would bring a relevant performance gain in terms of level or stability (this hypothesis failed), we decided for the standard SM [VWC %] as the most suitable concept for the ML target variable.

Fig. 5 We also realize that it is significantly more challenging to forecast maximum than minimum SM (RMSE is 2.5 to 3 times greater). This is because maximum SM tends to be disturbed more aggressively by water intake, generating sharp SM peaks[6]. And as there is high uncertainty about water intake[7], this is reflected in greater difficulty in forecasting. With minimum SM, its behavior is more forecastable. First, because, being a minimum value, it does not consider those SM peaks that are difficult to forecast, being less affected by water intake events. Second, when water intake occurs, SM tends to quickly seek stabilization at lower levels, close to the wilting point (WP)[8], causing minimum SM to stay at low values with low variance. It stabilizes because the drier the soil, the more inaccessible it is for plants' transpiration processes, in addition to less water evaporated. Thus, the SM level changes slightly.

6.2. Feature group selection

When running our adapted forward stepwise for all the proposed scenarios (section 5.5), we can conclude that, in general, machine learning-based soil moisture (SM) forecast depends mostly on historic SM and soil matric potential (SMP), precipitation forecast, and a context-awareness index-based exclusively on sensing data (our ACI – abnormal context index – section 5.3). Thus, a smart irrigation solution designer can focus their data sourcing efforts on these data. We think this is an important finding that comes with a side-effect: machine learning-based SM forecast does not depend on traditional domain-knowledge features as evapotranspiration (ET), crop phenology, and soil hydraulic behavior models, i.e., as the IoT consolidates itself as a mainstream practical approach in agriculture, we speculate that there is a trend in favor of full data-driven approaches and a noticeable decrease in the importance of agronomic models in smart irrigation. We can say the full data approach is made possible by our context-awareness feature-engineered ACI, which implicitly captures domain-knowledge information. Without the ACI, domain knowledge features cannot be removed without losing relevant accuracy levels.

Table 7 brings the feature group selection results for the proposed scenarios (a, b, c, and d). For each of them, there are the feature groups added in each iteration of the adaptable forward stepwise method, with the respective values of R^2_{adj} of the resulting model and the $\Delta\left(R^2_{adj}\right)$ related to the resulting model from the previous iteration. Feature groups marked with * are those we consider to be of high relevance, whose criterion we use is $\Delta\left(R^2_{adj}\right) \geq 0.005$. On the other hand, feature groups marked with ** are those we consider of low but positive relevance ($0.000 \leq \Delta\left(R^2_{adj}\right) < 0.005$). We can confirm in Table 7 that even considering the low but positive relevance feature groups, there are no domain-knowledge engineered features.

For minimum SM forecast scenarios (Table 7 – a, b), only current SM and SMP data (from the three soil depths) are sufficient for near-full accuracy. Considering non-weighted training and metric type (a), five-days SM historic and ACI provide marginal accuracy improvements. Considering weighted training and metric type (b), historic precipitation also takes part in a marginal accuracy improvement.

For maximum SM forecast scenarios (Table 7 – c, d), current SM and SMP (from the three soil depths) and ACI data are most responsible for accuracy. In the case of weighted training and metric type (d), the ACI increases in relevance, and the precipitation forecast provides a sharp difference in the accuracy. Such behavior makes sense since the weighted setup privileges critical data points whose predictions are more complicated (require more data for better contextual reading) and related to precipitation uncertainties.

[6] See actual soil moisture (SM) behavior reacting to water intake (the moments when SM raises) in the first graph of Fig. 8 of section 6.4.

[7] Precipitation forecast and probability of precipitation are naturally uncertain (see the bottom graph of Fig. 8 of section 6.4).

[8] See actual soil moisture (SM) behavior reacting after SM peak due to water intake (the moments when SM readily decreases) in the first graph of Fig. 8 of section 6.4. The more water available, the more accessible and easier it is to consume (Van Genuchten, 1980; Rawls et al., 1991), the faster the SM level decreases.

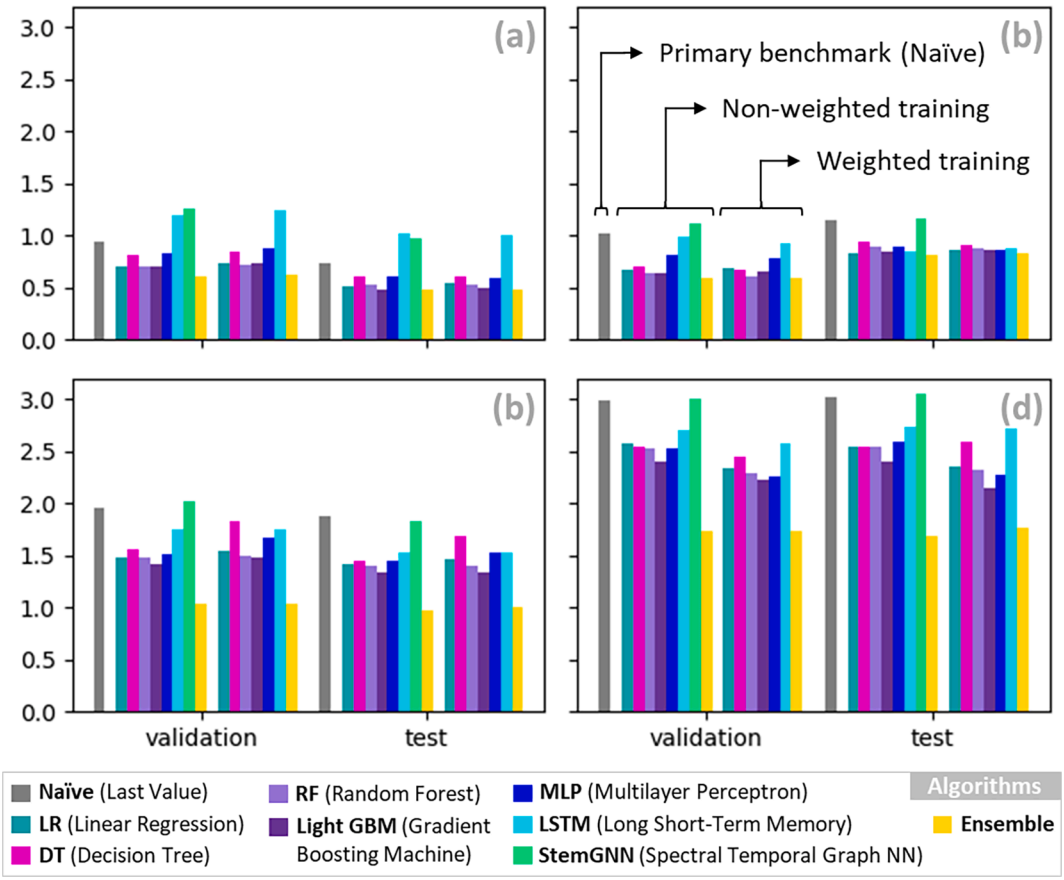

Fig. 6. Algorithm comparison – RMSE [SM[VWC %]] for D + 1 forecast. (a) Target: SM MIN, Metric weighting: NO; (b) Target: SM MIN, Metric weighting: YES; (c) Target: SM MAX, Metric weighting: NO; (d) Target: SM MAX, Metric weighting: YES.

We decided to choose all feature groups shown in Table 7, even those with low but positive relevance. The other feature groups are disregarded for the following tasks (algorithm comparison and ensemble - sections 6.3 and 6.4, respectively). Thus, the algorithms tested in algorithm comparison (section 6.3) can look for greater relevance in feature groups classified here as low but positive relevance.

6.3. Algorithms comparison

Fig. 6 shows the comparison among the machine learning (ML) algorithms, considering RMSE [SM[VWC %]] as metric and D + 1 as forecast horizon. Four graphs are represented for each combination of the target variable (minimum and maximum SM) and metric type (non-weighted and weighted calculated). In each graph, the accuracy is presented for the validation and testing datasets, allowing the evaluation of accuracy stability between these datasets. Note that for each evaluated dataset (validation and test), there are three column beams: the first contains a single gray column, with our naïve approach accuracy; the second beam contains the accuracies of models resulting from non-weighted training; the third beam contains the accuracies of the models resulting from weighted training. Note that the StemGNN algorithm has no representative model for weighted training. This is because the implementation of this algorithm does not consider the possibility of weighted training (Cao et al., 2020; Microsoft, 2021b).

Among the competing standalone algorithms (RL, DT, RF, LightGBM, LSTM, and StemGNN), LightGBM is the one that has the best performance across all the scenarios. Moreover, its accuracy is relatively stable across all the scenarios. The scenarios are any combination of validation vs. test, non-weighted vs. weighted training, and non-weighted vs. weighted metric type (Fig. 6). LightGBM advantage is salient when critical data points are weighted, either via training (weighted training), via metric (weighted metric type), or even via target variable (maximum SM, as it is more sensitive to critical data points). This makes us infer that the sequential predictors of gradient boosting machines, which LightGBM represents, are capturing traits of critical data points while maintaining performance in other data points.

Algorithm ensemble proved to be a smart strategy, especially for the maximum SM forecast, reducing the RMSE between 20 and 30% in all scenarios compared to LightGBM, the best standalone algorithm (Fig. 6). This shows that isolated algorithms capture different behavioral characteristics and that the ensemble manages to capture the best of each (Géron, 2019; Goodfellow et al., 2015). And, given the magnitude of accuracy improvement, this behavior may be generalizable to other real cases. As for the minimum SM forecast, the RMSE reduction was smaller (0 to 8%) but still relevant.

Recurrence-based time-series algorithms (represented by LSTM and StemGNN) performed relatively poorly (Fig. 6). This is probably due to the many discontinuities in the SM and SMP data from soil probes, subject to equipment failures (of sensing and in-field communication), given that these algorithms' functional structure assumes data continuity (Géron, 2019; Goodfellow et al., 2015). However, with the development of more robust IoT solutions (Junior & Kamienski, 2021; Togneri et al., 2019) and the deployment of data sourcing redundancy [9](Ardagna, Cappiello, Samá, & Vitali, 2018; Togneri et al., 2019), data quality is expected to improve. This is likely to bring performance improvement of recurrence-based time-series algorithms.

[9] E.g., SM being collected by redundant soil probes, or by both soil probes and remote sensing.

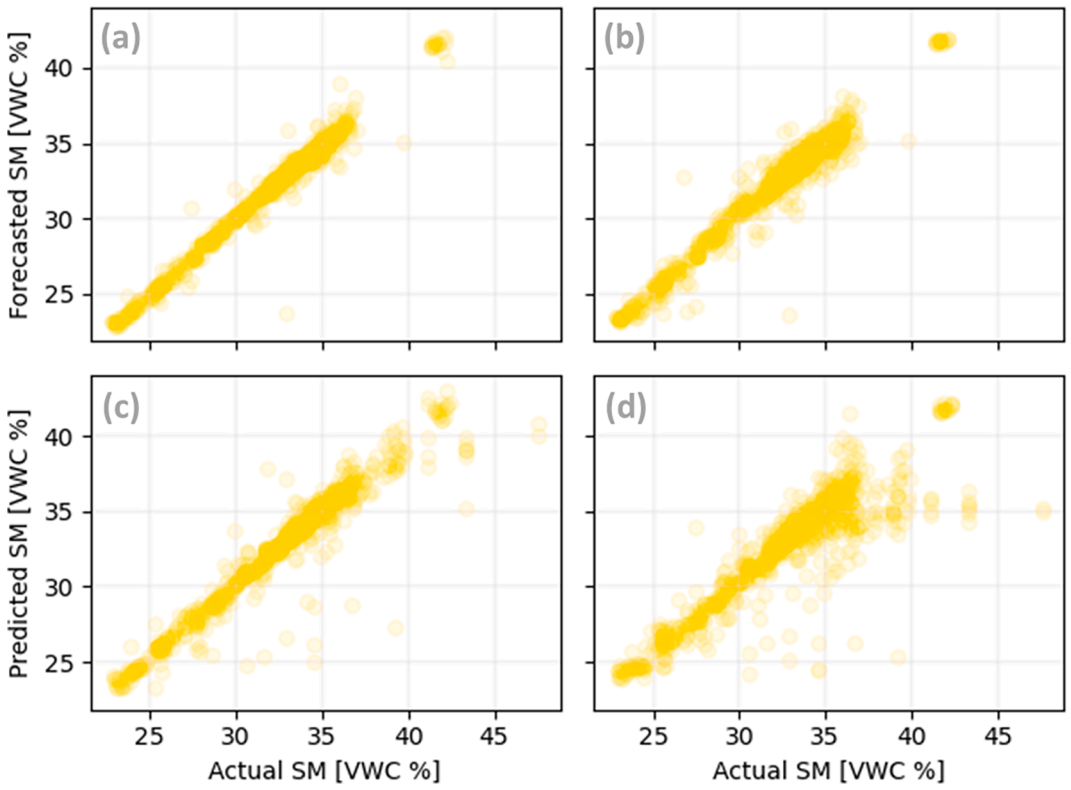

Fig. 7. Evaluation of non-weighted trained ensemble – Actual vs. Forecasted SM in the test dataset. (a) Target: MIN in D + 1; (b) Target: MIN in D + 2; (c) Target: MAX in D + 1; (d) Target: MAX in D + 2.

Weighted training only performed significantly higher (3 to 9%) for the following conditions: maximum SM as target variable, weighted metric type, and considering the standalone algorithms (Fig. 6), all taken together. Even limited to some scenarios, this result validates the hypothesis that weighted training increases accuracy for critical data points while maintaining accuracy for typical data points. This is already useful for solution designers who want to use a single ML algorithm for SM forecast. However, we realize that non-weighted or weighted training makes no difference for the algorithm ensemble: the results are equivalent. But this is not to say that weighted training has lost its usefulness in algorithm ensemble. In any case, we should note that our ensemble considers all non-weighted and weighted trained model versions (section 5.7), i.e., ensembles are implicitly benefited from weighted training.

For the metric type, we can learn that it is always valid to measure the results in both non-weighted and weighted setups. With the non-weighted metric type, we estimate the overall accuracy. With weighted metric type, we become aware of the accuracy gap between typical days and days of more significant weather uncertainty (our critical data points).

Table A-4 (for D + 1) and **Table A-5** (for D + 2) in the Appendix bring the complete set of results for the test dataset, considering a wider range of metrics: RMSE [SM[VWC %]], MAE [SM[VWC %]], R^2_{adj}, and MAPE. However, for all the metrics, the insights are similar to those shown by Fig. 6, and the tables can be accessed in the reader's interest for further details.

In summary, algorithm comparison main insights are: (i) if the data has discontinuities and if only a single ML algorithm is used, we recommend using decision tree-based ensembles, LightGBM in particular; (ii) an algorithm ensemble, using both weighted and non-weighted training standalone algorithms, can improve the accuracy by up to 20 to 30%, and therefore should be considered; (iii) we recommend weighted training because it improves accuracy on critical days without worsening accuracy on typical days; (iv) we recommend measuring performance by both non-weighted and weighted metric type. The tuned hyperparameters of the resulting models are found in table **Table A-3** in Appendix.

6.4. *Ensemble performance*

In section 6.3, we notice that our algorithm ensemble provides 20 to 30% of RMSE improvement for the maximum soil moisture (SM) forecast and 0 to 8% for the minimum SM forecast, which validates the hypothesis that an algorithm ensemble is a good option. We exploit our algorithm ensemble in the current section, as it is our maximum expression of SM forecast performance.

Fig. 7 shows actual vs. forecasted values in the test dataset. We use non-weighted trained model versions as they showed to be the adequate option according to section 6.3. There are four graphs in the figure, each of them refers to a combined target variable (minimum vs. maximum SM) and forecast horizon (D + 1 vs. D + 2) scenarios. Note that the occurrences concentration is on the line forecasted SM = current SM in all scenarios, which is expected. We can also see that the variance of points for maximum SM is greater than for minimum SM, confirming that it is more challenging to forecast maximum values because they are more susceptible to precipitation uncertainties (critical days). Another expected perception that we can confirm is a greater variance for D + 2 than for D + 1, indicating that it is more challenging to forecast two days than one day ahead.

Table 8
Ensemble evaluation – All metrics in the test dataset.

			Target: SM MIN				Target: SM MAX			
Horizon	Metric weighting	Training weighting	RMSE SM [VWC %]	MAE SM [VWC %]	R^2_{adj}	MAPE	RMSE SM [VWC %]	MAE SM [VWC %]	R^2_{adj}	MAPE
D + 1	**NO**	**NO**	0.48	0.24	0.98	0.7%	0.97	0.43	0.93	1.3%
		YES	0.48	0.23	0.98	0.7%	1.00	0.47	0.93	1.4%
	YES	**NO**	0.82	0.33	0.95	1.0%	1.69	0.83	0.86	2.4%
		YES	0.83	0.33	0.95	1.0%	1.76	0.87	0.85	2.5%
D + 2	**NO**	**NO**	0.67	0.39	0.96	1.2%	1.66	0.94	0.80	2.8%
		YES	0.72	0.45	0.96	1.4%	1.66	0.93	0.80	2.7%
	YES	**NO**	0.87	0.57	0.94	1.8%	1.73	1.08	0.81	3.2%
		YES	0.86	0.58	0.95	1.8%	1.73	1.07	0.81	3.1%

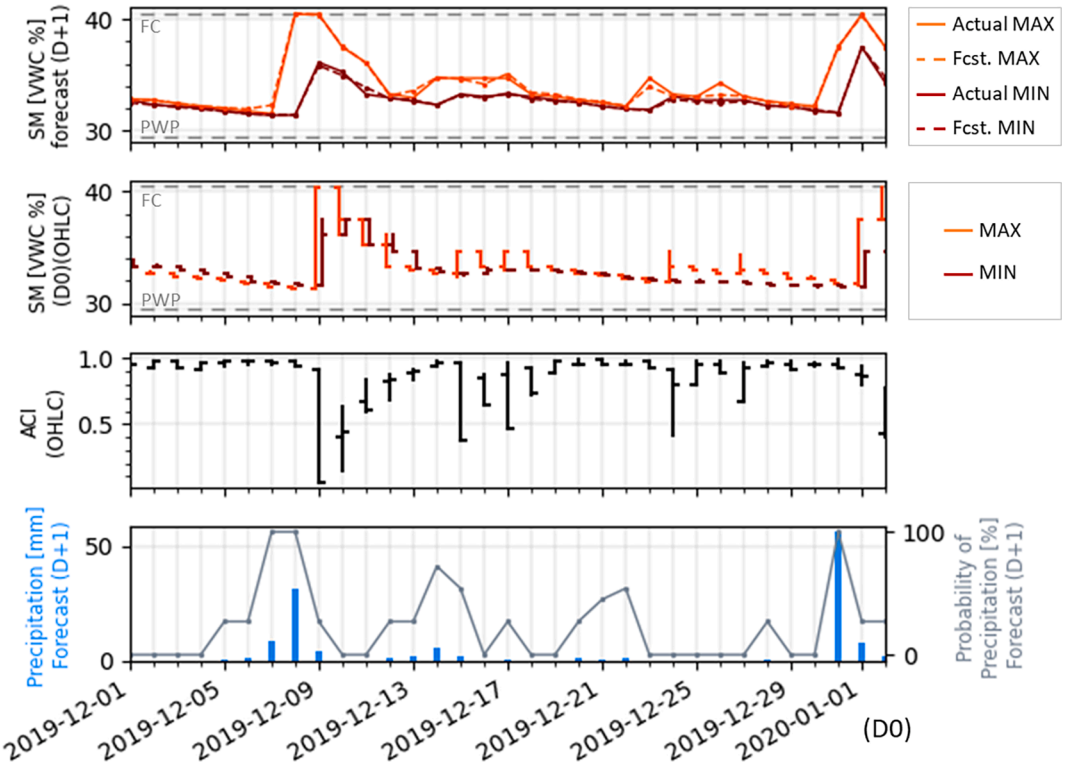

Fig. 8. Evaluation of non-weighted trained ensemble – Example of timeline of D + 1 SM forecasting and dependency analysis in the test dataset (farm 1 – field 1).

Table 8 compiles algorithm ensemble accuracies into test datasets for all modeling scenarios. In addition to the RMSE [SM[VWC %]], MAE [SM[VWC %]], R^2_{adj}, and MAPE are being considered. Considering non-weighted training and non-weighted metric type, we can observe that, for D + 1 minimum SM, $R^2_{adj} = 0.98$ and MAPE = 0.7%, and for D + 1 maximum SM, $R^2_{adj} = 0.93$ and MAPE = 1.3%. We are considering a non-weighted metric type so that our results are comparable to the related work references (section 2.1), which use this metric type. And we are using R^2_{adj} and MAPE for comparison with those references as these metrics are independent of the target variable chosen concept, allowing the relativization of performance across all references. Our results cannot be directly compared with section 2.1 references, as they use a single target variable which value is measured at a fixed day time. However, we can assume that if we used the same definition of the target variable, our performance would come closer to minimum SM values, as there is always a tendency to concentrate on lower values and to quickly return to lower values after peaks (this behavior can be realized in Fig. 6 first and second graphs). Taking the accuracy to minimum SM as a reference ($R^2_{adj} = 0.98$ and MAPE = 0.7%), we can see that our algorithm ensemble has similar accuracy levels to the related work references (section 2.1). Rather than concluding that our approach performs comparable to current best practices, we can advocate that, when working with non-central tendency measures (minimum and maximum SM), we favor greater awareness of the extreme levels of SM that can happen during the daytime to facilitate mitigation of water stress or leaching risks. Also, in Table 8, following the same criteria (non-

weighted training, non-weighted metric type), we collected $R^2_{adj} = 0.96$ and MAPE = 1.2% for D + 2 minimum SM, and $R^2_{adj} = 0.80$ and MAPE = 2.8% for D + 2 maximum SM. Although D + 2 values are worse than those of D + 1, which is expected, they also have similar levels to those observed in the related work references (section 2.1) for D + 2.

Fig. 8 illustrates a sample of SM forecasting in a time scale and features, allowing the intuition about the relationship among the relevant factors. In section 6.2, we found out that SM forecast relies mainly on SM, abnormal context index (ACI), and precipitation forecast data. The figure has four graphs: the one at the top contains D + 1 actual and forecasted values of minimum and maximum SM; the second graph contains D + 0 minimum and maximum SM intraday values, arranged in OHLC representation[10]; the third graph contains ACI intraday values, also in OHLC representation; and the bottom graph has precipitation and probability of precipitation forecasts for D + 1.

Observe that in Fig. 8, until approximately 2019–12-06, in zero or low precipitation values and probability of precipitation forecasts, actual SM continually and smoothly decreases by a soft and near-constant delta. In 2019–12-09 as D0, we have the first relevant positive delta of SM to predict for D + 1. Note that SM in D0 remains at low levels and ACI at high levels (in the case of ACI, it means a typical multivariate behavior), not indicating that SM would change for D + 1. However, the probability of precipitation forecast is 100%, and the precipitation forecast is 25mm [11], a considerable volume of water intake. This was enough to forecast that SM would go to 40%*VWC* for D + 1. In 2019–12-10 as D0, both precipitation and probability of precipitation forecasts are low, indicating that SM should decrease to D + 1, but that is not what happened. Minimum SM increased, and maximum SM remained in the field capacity (FC). Such counter-intuitive behavior occurred due to the combination of SM values in D0, which were high due to the precipitation that happened on that day, and ACI, which was low, indicating an unusual multivariate condition. In this case, we found that this abnormal condition was water saturation in deeper soil layers, a relatively rare condition occurring in the rainiest periods of the year and after relevant precipitation. With high SM levels and the deeper soil layers filled, the soil remained at high SM levels for another day. From 2019 to 12-11 onwards, fewer precipitation effects and the return of ACI to higher levels allow the forecast to keep up with the continuous and smooth decrease (with some occasional variations) until 2019–12-31 when the high values of precipitation and probability of precipitation forecasts again allowed to predict a new positive delta of SM in D + 1. If we cannot anticipate mainly these short-term precipitation events, crops would face a significant risk. The irrigation system could have applied water to the soil. The combined effect of irrigation and precipitation would provoke leaching (SM being near field capacity (FC) for more extended periods), carrying soil nutrients to deeper soil layers.

The capacity of correctly predicting these situations is a case in point of the important value that our proposed setup can deliver. In other words, this D + 1 prediction could call forth actions to save resources and to conduct proactive measures.

7. Conclusions

The rise of the Internet of Things allowed higher spatial–temporal resolution soil moisture data captured through in situ sensing. Such abundance of data enables machine learning-based soil moisture (SM) forecast as an alternative to traditional mechanistic approaches for irrigation water need estimation. The general objective of this paper is to improve machine learning-based SM forecasting for smart irrigation applications. By running the methodology, we could establish answers to our research question organized by topics:

(i) Target variable definition: Non-central tendency measures are presumably more useful than central ones or any single value of a specific daytime. They allow better characterizing the distribution of probabilities and extreme events that represent some risk to plant development. This paper specifically shows the usefulness of using minimum and maximum SM as target variables (sections 5.4 and 6.1). Furthermore, SM [VWC %] is the best concept for target variable against SMP [kPa] for ML-based SM forecast (section 6.1).

(ii) Context-awareness importance: multidimensional anomaly detection monitoring plays a relevant role based on sensing data (of SM and weather). It captures latent behavior from data, especially those of rare but relevant states (section 6.2) and domain-knowledge (section 5.3). Our context-aware index is one of the important factors for ML-based SM forecast, even enabling a full data-driven approach. Moreover, it made a full data-driven approach possible, as it implicitly captured domain-knowledge feature behaviour. Finally, we suggest our intraday autoencoder approach (section 5.3) as a starting point for other implementations.

(iii) Feature group selection: ML-based SM forecasting reaches its maximum performance relying only on SM and SMP[12], multi-sensor context-awareness, and precipitation forecast data (section 6.2). Furthermore, ML-based SM forecasting is independent of traditional domain-knowledge features as physical-empirical models for evapotranspiration (ET), crop phenology, and soil (section 6.2). This indicates that if IoT consolidates as a robust and viable solution, there is an alternative to agronomy domain-knowledge.

(iv) Algorithms comparison: If using a standalone algorithm, we recommend using LightGBM as a starting point, as it reached the best and most stable performance among all alternatives we have tested. Additionally, in general, among state-of-the-art algorithms for time-series problems, decision tree ensembles outperformed recurrent neural networks (section 6.3).

(v) Algorithm ensemble utility: Blending predictions via algorithm ensemble provides an additional gain in accuracy, mainly for maximum SM forecast (section 6.4).

(vi) Dataset weighting utility for model training and measuring performance: Weighted training brings forecast accuracy gains to critical data points (days that are more difficult to forecast due to precipitation uncertainties) without losing accuracy on typical data points forecast (sections 6.1 to 6.4). Furthermore, it is helpful to measure accuracy both with non-weighted and weighted metrics. The former allows for measuring overall accuracy, and the latter provides for consciousness over how poorer the accuracy for critical data points are (sections 6.1 to 6.4). An alternative could be to measure accuracy separately for critical data points and the rest.

Data quality insights emerged as a side-effect of our investigation. The robustness of the sensing solutions is a point of attention when choosing to implement ML approaches to SM forecast. ML approaches only work when data is available and reliable (Géron, 2019; Goodfellow et al., 2015). This is an issue mainly for SM coming from soil probes, which are more subject to interruption and decalibration because they are subject to bad weather, the risk of damage by agricultural machinery, and because they are difficult to access for maintenance (often soil probes are located amidst dense crops of multi-kilometric dimensions)

[10] OHLC (open, high, low, close) is a typical graphical representation for asset price variation in a financial market that allows the representation of statistics in time windows: open, to the first recorded value; high, to the highest recorded value; low, for the lowest recorded value; and close, to the last recorded value (Rockefeller, 2019).

[11] Section 2.7 explains how water volume can be measured in mm.

[12] From three depths in our case: 20 cm, 40 cm, and 60 cm.

(Kamienski et al., 2019). Therefore, we recommend data sourcing to be redundant, either by same-type sensor redundancy (Junior & Kamienski, 2021; Moon, Kim, Zhang, & Son, 2018; Torres, da Rocha, da Silva, de Souza, & Gondim, 2020) or by multiple data sourcing (Ardagna et al., 2018; Togneri et al., 2019) for each monitored feature. Combining in situ and remote sensing simultaneously is an example of multiple data sourcing. If sensing is not robust, we recommend deploying a traditional water need estimation (WNE) model like soil water balance (SWB) ones (Jensen & Allen, 2016), based on domain-knowledge models, to serve as a backup. Furthermore, robust sensing solutions can improve data quality and, therefore, improve recurrence-based algorithms' performance (Goodfellow et al., 2015), such as the LSTM and StemGNN that we tested. If it is the case of data with little or no discontinuity, this class of algorithms can be tested for better performance.

Our findings may generalize well, at least within the breadth of our tests and the diversity of our real case, which has twelve fields from four farms spread across different geographic and climatic zones of Brazil, working eight different crop types in 55 crops. For example, the climates covered in this work correspond to 25.5% of the world's area suitable for agriculture (section 4). Finally, we emphasize the importance of water need estimation (WNE) and SM forecast accuracy for smart irrigation schemes. The development of different systematic approaches aims to democratize smart irrigation for the greatest number of farmers and their specific conditions, to reduce operating costs (electricity tariff), mitigate crop yield loss risks, and save water as a it is a scarce natural resource (Kamienski et al., 2019).

8. Future work

This work opens the door for a series of next steps, divided into the following subjects: target variable definition, modeling, data quality, and performance benchmarking.

For the target variable definition, we suggest testing more non-central tendency measures beyond just minimum and maximum SM, e.g., percentiles. Such an alternative would allow the delineation of the future probabilistic distribution of SM. This test would require the ML algorithms to be adapted to act as quantile regressors. Another alternative that would allow the assessment of future uncertainties would be the use of stochastic simulation approaches (Cai et al., 2011; Linker et al., 2016; Liu, Li, Huang, Zhuang, & Fu, 2017; Shang et al., 2020).

For modeling, we suggest seven possibilities: (i) enhance time granularity by testing hourly instead of daily time step, both for feature aggregating and SM forecast, to verify whether it provides practical benefits for crops where continuous irrigation is a possibility; (ii) forecast SM for more than one soil depth (in our case, we only predicted SM for a depth of 20 cm, although we had and have used data from other depths), allowing forecasting studies on the vertical profile of SM; (iii) refine understanding of input data with a focus on individual features, and not only considering features groups as we do in this paper, being able to explore feature engineering and feature importance analysis (Chandrashekar & Sahin, 2014); despite being applied to a different context, an alternative feature learning and latent space encoding is given by (Pattathal, 2022); (iv) test alternative loss functions than RMSE (Géron, 2019; Goodfellow et al., 2015); (v) test alternative algorithms, as standalone approaches and for algorithm ensemble (Géron, 2019; Goodfellow et al., 2015; Liu, Gong, Yang, & Chen, 2020); (vi) test algorithm stacking, as an alternative to blending independent model predictions (Géron, 2019; Goodfellow et al., 2015); (vii) explore other context-awareness approaches, using or not anomaly detection techniques (Ardagna et al., 2018; Bertossi & Geerts, 2020; Erhan et al., 2021; Hariri, Fredericks, & Bowers, 2019; Srikanth, Branch, Jin, & Jugdutt, 2020).

For data quality, we identified three focuses that would help improve ML-based SM forecast performance: (i) study the effect of sensor redundancy or multiple data sourcing on data availability and reliability, and, consequently, in the feasibility of full data-driven approaches; we recommend start by reading (Ardagna et al., 2018; Junior & Kamienski, 2021; Moon et al., 2018; Togneri et al., 2019); moreover, we recommend Anton *et al.* (Anton, Matei, & Avram, 2019a, 2019b), that advocates multiple data sourcing can improve not only ML reliability but also ML accuracy on temperature and SM when data is already reliable; (ii) apply multiple time-series data imputation (Cao, Wang, Li, Zhou, Li, & Li, 2018; Liu et al., 2017; Luo, Cai, Zhang, Xu, & Yuan, 2018; Luo, Zhang, Cai, & Yua, 2019; Raubitzek & Neubauer, 2021; Singh, Deznabi, Narasimhan, Kucharski, Uppaal, Josyula, & Fiterau, 2019; Yildiz, Heinonen, & Lähdesmäki, 2019) to understand the potential of these techniques in alleviating the problem of data discontinuity, especially SM data; (iii) from the available sensor data, investigate how to separate anomalous but reliable data from unreliable data (e.g., sensor decalibration, frequent in soil probes), to increase sensing data reliability; we recommend start by using clustering and anomaly detection techniques (Amarbayasgalan et al., 2020; Cai, Li, Li, Sun, & Yuan, 2020; Chandola, Banerjee, & Kumar, 2009) in multidimensional sensor data and IoT telemetry.

Finally, for performance benchmarking, we encourage direct comparisons between different types of models of irrigation water need estimation (WNE): ML-based SM forecast vs. traditional soil water balance (SWB) (Gu et al., 2017; Jensen & Allen, 2016; Linker et al., 2016; Pereira, Paredes, & Jovanovic, 2020), preferably in real cases. Moreover, for ML-based SM forecast, it would also be interesting to see other real-world cases comparing models that use domain-knowledge features with full data-driven approaches.

CRediT authorship contribution statement

Rodrigo Togneri: Conceptualization, Methodology, Software, Validation, Formal analysis, Investigation, Data curation, Writing – original draft, Visualization, Supervision, Project administration. **Diego Felipe dos Santos:** Conceptualization, Resources, Data curation. **Glauber Camponogara:** Conceptualization, Resources, Data curation. **Hitoshi Nagano:** Methodology, Software, Investigation, Writing – review & editing, Supervision. **Gilliard Custódio:** Methodology, Software, Investigation. **Ronaldo Prati:** Writing – review & editing, Supervision. **Stênio Fernandes:** Methodology, Writing – review & editing, Supervision. **Carlos Kamienski:** Writing – review & editing, Supervision, Project administration.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

We would like to thank Agrosmart for providing data and useful insights on this work.

Appendix

Table A1
Feature groups (detailed version) - Options to SM forecast modeling.

Feature Class	Features	Description	Dimensional (quant.) /Categories (quali.)	Feature groupcandidates (blocks)		
				Lagged 5 days	Current (D0)	Forecast (2 days)
Weather(except precipitation)	T	Atmospheric temperature.	[C]	X	X	X
	Rad	Global solar radiation.	[W/m^2]			
	RH	Atmospheric relative humidity.	[%]			
	ws (2m)	Wind speed at 2 m high.	[m/s]			
Crop Phenologyand ET	ETo (FAO-56 Penman-Monteith)	Reference evapotranspiration calculated by the FAO-56 Penman-Monteith method.	[mm/m^2]			
	ETo (Hargreaves)	Reference evapotranspiration calculated by the Hargreaves method.				
	Kc	Crop coefficient, that stands to relate ETo to ETc depending on culture and crop phenological stage.	[]	X	X	X
	ETc (FAO-56 Penman-Monteith)	Crop evapotranspiration calculated by the FAO-56 Penman-Monteith method, considering ETo (FAO-56 Penman-Monteith) and Kc.	[mm/m^2]			
	ETc (Hargreaves)	Crop evapotranspiration calculated by the Hargreaves method, considering ETo (FAO-56 Hargreaves) and Kc.				
Water Intake	realized water intake	Realized water intake comprehending precipitation and irrigation.	[mm/m^2]	X	X	
	precipitation forecast	Total estimated amount of precipitation water.				X
	probability of precipitation	Probability of a precipitation event.	[%]			
SM, SMP	SM MIN,SMP MIN	(Soil moisture, Soil matric potential) - day minimum value.	SM [*% VWC*]SM [% VWC normalized] SMP [kPa]	X	X	Target
	SM MAX,SMP MAX	(Soil moisture, Soil matric potential) - day maximum value.				
	SM OPEN,SMP OPEN	(Soil moisture, Soil matric potential) - day open (first) value.				
	SM CLOSE,SMP CLOSE	(Soil moisture, Soil matric potential) - day close (last) value.				
	SM MEAN,SMP MEAN	(Soil moisture, Soil matric potential) - day mean value.				
	SM STD,SMP STD	(Soil moisture, Soil matric potential) - day standard deviation value.				
ACI (SM)	ACI (SM)	Abnormal context index (ACI) of SM features considering intraday behavior.	[]	X	X	
Soil	sd	Dry soil density.	[kg/m^3]			
	PWP	Permanent wilting point: the SM below which the plant wilts permanently and dies.	[% VWC] [% VWC normalized] [kPa]			
	SM(~ 100kPaof SWT)	SM corresponding to 100 kPa of soil water tension (SWT).			X	
	SM(~ 10kPaof SWT)	SM corresponding to 10 kPa of soil water tension (SWT).				
	FC	Field capacity: the maximum SM the soil can hold before allowing leaching.				
Location	(farm, field)	Farm and field considered together.	[(1, 1), (1, 2), (1, 3),(1, 4), (1, 5), (1, 6),(2, 1), (2, 2), (2, 3),(3, 1), (3, 2), (4, 1)]		X	
Time	day of the year	The day of the year.	[day]		X	
Crop Type	crop type	The vegetable planted in the field.	[soybeans, wheat, white oats, black oats, yellow maize, brown maize, coffee, pinto beans]		X	
	culture accumulated days	The age (in days) of a crop from a particular culture.	[*day*]			

Note 1: Daily time step (original sensor data in intraday).
Note 2: Imputation only occurred at time intervals with small unavailable data intervals, up to 5 h.

Table A2
Algorithms hyperparameter spaces.

Algorithm	Framework	Hyperparameters Space
LR	scikit-learn 1.0.1	fit_intercept = True
DT	scikit-learn 1.0.1	min_samples_leaf = (1, 40) max_depth = (1, 20)
RF	scikit-learn 1.0.1	min_samples_leaf = (1, 40) max_depth = (1, 20) n_estimators = (5, 1500)
Light GBM	lightgbm 3.3.1	num_leaves = X_train.shape[0] max_depth = (1,20) min_child_samples = (1,40) subsamble = (0.05,1.) colsample_bytree = (0.05,1.) bagging_freq = 1 n_estimators = (5,1500) learning_rate = (1e-3, 1e-1, 'log-uniform')
MLP	Keras (Tensorflow 2)	input_dimension = X_train.shape[1] hidden_layers = (3,10) neurons_per_hidden_layer = int(X_train.shape[1]*0.5), int(X_train.shape[1]*2.) dropout_percentage = (0.01,0.5) learning_rate = (1e-3, 1e-1, 'log-uniform') optimizer = adam
LSTM	Keras (Tensorflow 2)	convolutional_width = [2, 3] dropout_percentage = [0.1, 0.2, 0.3] patience = [2, 3, 4]
StemGNN	StemGNN	window_size = (1, 20) multi_layer = (2, 40)
Ensemble (Light GBM)	lightgbm 3.3.1	num_leaves = X_train.shape[0] max_depth = (1,20) min_child_samples = (1,40) subsamble = subsample colsample_bytree = (0.05,1.) bagging_freq = 1 n_estimators = (5,1500) learning_rate = (1e-3, 1e-1, 'log-uniform')

Note 1: () used for Interval of possible values; [] used for a set of possible values.
Note 2: All the non-cited hyperparameters are set to standard values.
Note 3: Scikit-learn (Scikit-Learn, 2021f), LightGBM (Microsoft, 2021a), Keras (Keras, 2021), StemGNN (Microsoft, 2021b).

Table A3
Algorithms tuned hyperparameters.

Algorithm	Training weighting	Target	Tuned Hyperparameters	
LR	NO	SM	D + 1	fit_intercept = True
		MIN	D + 2	fit_intercept = True
		SM	D + 1	fit_intercept = True
		MAX	D + 2	fit_intercept = True
	YES	SM	D + 1	fit_intercept = True
		MIN	D + 2	fit_intercept = True
		SM	D + 1	fit_intercept = True
		MAX	D + 2	fit_intercept = True
DT	NO	SM	D + 1	min_samples_leaf = 14, max_depth = 9
		MIN	D + 2	min_samples_leaf = 10, max_depth = 5
		SM	D + 1	min_samples_leaf = 11, max_depth = 9
		MAX	D + 2	min_samples_leaf = 34, max_depth = 4
	YES	SM	D + 1	min_samples_leaf = 31, max_depth = 6
		MIN	D + 2	min_samples_leaf = 33, max_depth = 4
		SM	D + 1	min_samples_leaf = 8, max_depth = 19
		MAX	D + 2	min_samples_leaf = 26, max_depth = 3
RF	NO	SM	D + 1	min_samples_leaf = 8, max_depth = 14, n_estimators = 917
		MIN	D + 2	min_samples_leaf = 5, max_depth = 6, n_estimators = 1039
		SM	D + 1	min_samples_leaf = 2, max_depth = 8, n_estimators = 46
		MAX	D + 2	min_samples_leaf = 5, max_depth = 12, n_estimators = 503
	YES	SM	D + 1	min_samples_leaf = 11, max_depth = 9, n_estimators = 177
		MIN	D + 2	min_samples_leaf = 40, max_depth = 5, n_estimators = 983
		SM	D + 1	min_samples_leaf = 8, max_depth = 14, n_estimators = 917
		MAX	D + 2	min_samples_leaf = 11, max_depth = 9, n_estimators = 177
Light GBM	NO	SM MIN	D + 1	num_leaves = X_train.shape[0], max_depth = 9, min_child_samples = 29, subsamble = 0.46, colsample_bytree = 0.49, bagging_freq = 1, n_estimators = 619, learning_rate = 0.00994

(continued on next page)

Table A3 (*continued*)

Algorithm	Training weighting	Target	Tuned Hyperparameters	
			D + 2	num_leaves = X_train.shape[0], max_depth = 9, min_child_samples = 40, subsamble = 0.58, colsample_bytree = 1.00, bagging_freq = 1, n_estimators = 1257, learning_rate = 0.00918
		SM MAX	D + 1	num_leaves = X_train.shape[0], max_depth = 4, min_child_samples = 33, subsamble = 0.60, colsample_bytree = 0.94, bagging_freq = 1, n_estimators = 580, learning_rate = 0.02699
			D + 2	num_leaves = X_train.shape[0], max_depth = 20, min_child_samples = 13, subsamble = 0.05, colsample_bytree = 1.00, bagging_freq = 1, n_estimators = 423, learning_rate = 0.01212
	YES	SM MIN	D + 1	num_leaves = X_train.shape[0], max_depth = 9, min_child_samples = 29, subsamble = 0.46, colsample_bytree = 0.49, bagging_freq = 1, n_estimators = 619, learning_rate = 0.00994
			D + 2	num_leaves = X_train.shape[0], max_depth = 9, min_child_samples = 40, subsamble = 0.58, colsample_bytree = 1.00, bagging_freq = 1, n_estimators = 1257, learning_rate = 0.00918
		SM MAX	D + 1	num_leaves = X_train.shape[0], max_depth = 4, min_child_samples = 33, subsamble = 0.60, colsample_bytree = 0.94, bagging_freq = 1, n_estimators = 580, learning_rate = 0.02699
			D + 2	num_leaves = X_train.shape[0], max_depth = 20, min_child_samples = 13, subsamble = 0.05, colsample_bytree = 1.00, bagging_freq = 1, n_estimators = 423, learning_rate = 0.01212
MLP	NO	SM MIN	D + 1	hidden_layers = 6, neurons_per_hidden_layer = int(X_train.shape[1]*0.72), dropout_percentage = 0.04, learning_rate = 0.00917
			D + 2	hidden_layers = 6, neurons_per_hidden_layer = int(X_train.shape[1]*0.64), dropout_percentage = 0.06, learning_rate = 0.00987
		SM MAX	D + 1	hidden_layers = 8, neurons_per_hidden_layer = int(X_train.shape[1]*0.98), dropout_percentage = 0.06, learning_rate = 0.02111
			D + 2	hidden_layers = 8, neurons_per_hidden_layer = int(X_train.shape[1]*1.04), dropout_percentage = 0.07, learning_rate = 0.01312
	YES	SM MIN	D + 1	hidden_layers = 6, neurons_per_hidden_layer = int(X_train.shape[1]*0.69), dropout_percentage = 0.04, learning_rate = 0.00992
			D + 2	hidden_layers = 7, neurons_per_hidden_layer = int(X_train.shape[1]*0.60), dropout_percentage = 0.05, learning_rate = 0.00921
		SM MAX	D + 1	hidden_layers = 8, neurons_per_hidden_layer = int(X_train.shape[1]*0.93), dropout_percentage = 0.07, learning_rate = 0.01915
			D + 2	hidden_layers = 9, neurons_per_hidden_layer = int(X_train.shape[1]*1.10), dropout_percentage = 0.07, learning_rate = 0.01456
LSTM	NO	SM	D + 1	dropout_percentage = 0.2, patience = 3, convolutional_width = 3
		MIN	D + 2	dropout_percentage = 0.1, patience = 2,convolutional_width = 2
		SM	D + 1	dropout_percentage = 0.1, patience = 2, convolutional_width = 3
		MAX	D + 2	dropout_percentage = 0.1, patience = 3, convolutional_width = 2
	YES	SM	D + 1	dropout_percentage = 0.2, patience = 2, convolutional_width = 3
		MIN	D + 2	dropout_percentage = 0.3, patience = 4, convolutional_width = 2
		SM	D + 1	dropout_percentage = 0.1, patience = 2, convolutional_width = 2
		MAX	D + 2	dropout_percentage = 0.3, patience = 2, convolutional_width = 2
StemGNN	NO	SM	D + 1	window_size = 6, multi_layer = 15
		MIN	D + 2	window_size = 8, multi_layer = 15
		SM	D + 1	window_size = 3, multi_layer = 20
		MAX	D + 2	window_size = 5, multi_layer = 10
Ensemble	NO	SM MIN	D + 1	num_leaves = X_train.shape[0], max_depth = 7, min_child_samples = 40, subsamble = 1.00, colsample_bytree = 1.00, bagging_freq = 1, n_estimators = 1500, learning_rate = 0.03178
			D + 2	num_leaves = X_train.shape[0], max_depth = 3, min_child_samples = 22, subsamble = 1.00, colsample_bytree = 0.10, bagging_freq = 1, n_estimators = 1289, learning_rate = 0.01283
		SM MAX	D + 1	num_leaves = X_train.shape[0], max_depth = 8, min_child_samples = 5, subsamble = 0.14, colsample_bytree = 1.00, bagging_freq = 1, n_estimators = 1449, learning_rate = 0.01575
			D + 2	num_leaves = X_train.shape[0], max_depth = 20, min_child_samples = 1, subsamble = 1.00, colsample_bytree = 0.14, bagging_freq = 1, n_estimators = 448, learning_rate = 0.01101
	YES	SM MIN	D + 1	num_leaves = X_train.shape[0], max_depth = 7, min_child_samples = 40, subsamble = 1.00, colsample_bytree = 1.00, bagging_freq = 1, n_estimators = 1500, learning_rate = 0.03178
			D + 2	num_leaves = X_train.shape[0], max_depth = 3, min_child_samples = 22, subsamble = 1.00, colsample_bytree = 0.10, bagging_freq = 1, n_estimators = 1289, learning_rate = 0.01283
		SM MAX	D + 1	num_leaves = X_train.shape[0], max_depth = 8, min_child_samples = 5, subsamble = 0.14, colsample_bytree = 1.00, bagging_freq = 1, n_estimators = 1449, learning_rate = 0.01575
			D + 2	num_leaves = X_train.shape[0], max_depth = 20, min_child_samples = 1, subsamble = 1.00, colsample_bytree = 0.14, bagging_freq = 1, n_estimators = 448, learning_rate = 0.01101

Table A4
Algorithm comparison – All metrics for D + 1 forecast in the test dataset.

			Target: SM MIN				Target: SM MAX			
			RMSE [SM[VWC %]]	MAE [SM[VWC %]]	R^2_{adj}	MAPE	RMSE [SM[VWC %]]	MAE [SM[VWC %]]	R^2_{adj}	MAPE
Metric weighting: NO	Naïve		0.74	0.32	0.95	1.0%	1.88	0.83	0.75	2.4%
	Training weighting: NO	LR	0.52	0.24	0.98	0.8%	1.42	0.64	0.86	1.8%
		DT	0.62	0.30	0.97	0.9%	1.45	0.68	0.85	2.0%
		RF	0.53	0.23	0.98	0.7%	1.41	0.67	0.86	1.9%
		Light GBM	0.48	0.24	0.98	0.7%	1.33	0.61	0.88	1.7%
		MLP	0.62	0.39	0.97	1.2%	1.46	0.72	0.85	2.1%
		LSTM	1.02	0.59	0.91	1.9%	1.53	0.75	0.84	2.2%
		StemGNN	0.98	0.49	0.92	1.6%	1.83	0.92	0.77	2.8%
		Ensemble	0.48	0.24	0.98	0.7%	0.97	0.43	0.93	1.3%
	Training weighting: YES	LR	0.55	0.26	0.97	0.8%	1.46	0.86	0.85	2.5%
		DT	0.61	0.32	0.97	1.0%	1.68	0.86	0.80	2.6%
		RF	0.53	0.23	0.98	0.7%	1.40	0.78	0.86	2.3%
		Light GBM	0.50	0.25	0.98	0.8%	1.34	0.75	0.88	2.2%
		MLP	0.59	0.39	0.97	1.2%	1.53	1.06	0.84	3.2%
		LSTM	1.01	0.58	0.92	1.8%	1.53	0.79	0.84	2.3%
		Ensemble	0.48	0.23	0.98	0.7%	1.00	0.47	0.93	1.4%
Metric weighting: YES	Naïve		1.15	0.46	0.90	1.4%	3.02	1.76	0.56	4.8%
	Training weighting: NO	LR	0.84	0.32	0.95	1.0%	2.54	1.47	0.68	4.0%
		DT	0.95	0.40	0.93	1.2%	2.53	1.49	0.68	4.1%
		RF	0.90	0.34	0.94	1.0%	2.54	1.50	0.68	4.1%
		Light GBM	0.84	0.34	0.95	1.1%	2.40	1.39	0.72	3.8%
		MLP	0.90	0.48	0.94	1.5%	2.59	1.55	0.67	4.3%
		LSTM	0.84	0.52	0.95	1.7%	2.74	1.70	0.60	4.7%
		StemGNN	1.16	0.61	0.90	1.9%	3.04	1.89	0.54	5.2%
		Ensemble	0.82	0.33	0.95	1.0%	1.69	0.83	0.86	2.4%
	Training weighting: YES	LR	0.86	0.35	0.94	1.1%	2.35	1.44	0.73	4.0%
		DT	0.92	0.39	0.94	1.2%	2.59	1.57	0.67	4.4%
		RF	0.88	0.33	0.94	1.0%	2.31	1.37	0.74	3.8%
		Light GBM	0.86	0.35	0.95	1.1%	2.15	1.30	0.77	3.6%
		MLP	0.86	0.46	0.94	1.4%	2.27	1.50	0.75	4.3%
		LSTM	0.87	0.56	0.94	1.8%	2.72	1.64	0.64	4.6%
		Ensemble	0.83	0.33	0.95	1.0%	1.76	0.87	0.85	2.5%

Table A5
Algorithm comparison – All metrics for D + 2 forecast in the test dataset.

			Target: SM MIN				Target: SM MAX			
			RMSE [SM[VWC %]]	MAE [SM[VWC %]]	R^2_{adj}	MAPE	RMSE [SM[VWC %]]	MAE [SM[VWC %]]	R^2_{adj}	MAPE
Metric weighting: NO	Naïve		0.89	0.49	0.93	1.5%	2.13	1.12	0.68	3.3%
	Training weighting: NO	LR	0.75	0.47	0.95	1.5%	1.72	0.93	0.79	2.7%
		DT	0.83	0.54	0.94	1.7%	1.84	1.14	0.76	3.4%
		RF	0.74	0.45	0.95	1.4%	1.71	0.97	0.80	2.9%
		Light GBM	0.80	0.53	0.95	1.7%	1.70	0.93	0.80	2.7%
		MLP	0.81	0.55	0.94	1.7%	1.70	0.99	0.80	3.0%
		LSTM	1.40	0.90	0.84	2.9%	1.96	1.13	0.73	3.4%
		StemGNN	1.06	0.59	0.91	1.9%	1.82	1.01	0.77	3.0%
		Ensemble	0.67	0.39	0.96	1.2%	1.66	0.94	0.80	2.8%
	Training weighting: YES	LR	0.86	0.61	0.94	1.9%	1.83	1.12	0.77	3.3%
		DT	0.98	0.70	0.92	2.2%	2.16	1.61	0.67	4.8%
		RF	0.87	0.61	0.94	1.9%	1.83	1.21	0.77	3.7%
		Light GBM	0.88	0.62	0.93	1.9%	1.75	1.04	0.79	3.1%
		MLP	0.94	0.69	0.93	2.2%	1.79	1.11	0.78	3.3%
		LSTM	1.41	0.87	0.84	2.8%	2.02	1.10	0.72	3.3%
		Ensemble	0.72	0.45	0.96	1.4%	1.66	0.93	0.80	2.7%
Metric weighting: YES	Naïve		1.19	0.79	0.89	2.4%	2.24	1.42	0.68	4.1%
	Training weighting: NO	LR	0.97	0.66	0.93	2.0%	1.83	1.12	0.79	3.2%
		DT	1.01	0.72	0.92	2.2%	2.01	1.33	0.75	3.9%
		RF	1.00	0.68	0.93	2.1%	1.86	1.18	0.78	3.4%
		Light GBM	0.99	0.72	0.93	2.2%	1.84	1.17	0.79	3.4%
		MLP	1.05	0.75	0.92	2.3%	1.80	1.13	0.80	3.3%
		LSTM	1.32	0.89	0.87	2.8%	1.85	1.18	0.79	3.6%
		StemGNN	1.14	0.76	0.90	2.4%	1.86	1.19	0.78	3.6%
		Ensemble	0.87	0.57	0.94	1.8%	1.73	1.08	0.81	3.2%
	Training weighting: YES	LR	0.95	0.68	0.93	2.1%	1.87	1.25	0.78	3.6%
		DT	1.09	0.81	0.91	2.5%	2.15	1.60	0.71	4.8%
		RF	0.99	0.71	0.93	2.2%	1.85	1.27	0.79	3.8%
		Light GBM	0.98	0.72	0.93	2.2%	1.85	1.24	0.78	3.7%
		MLP	1.02	0.77	0.92	2.4%	1.91	1.28	0.77	3.8%
		LSTM	1.32	0.85	0.87	2.7%	2.16	1.61	0.70	4.8%
		Ensemble	0.86	0.58	0.95	1.8%	1.73	1.07	0.81	3.1%

References

Abowarda, A., Bai, L., Zhang, C., Long, D., Li, X., Huang, Q., & Sun, Z. (2021). Generating surface soil moisture at 30 m spatial resolution using both data fusion and machine learning toward better water resources management at the field scale. *Remote Sensing of Environment, 255*, Article 112301. https://doi.org/10.1016/j.rse.2021.112301

Abrishambaf, O., Faria, P., Gomes, L., & Vale, Z. (2020). Agricultural irrigation scheduling for a crop management system considering water and energy use optimization. *Energy Reports, 6*(1), 133–139. https://doi.org/10.1016/j.egyr.2019.08.031

Adeyemi, O., Grove, I., Peets, S., Domun, Y., & Norton, T. (2018). Dynamic neural network modelling of soil moisture content for predictive irrigation scheduling. *Sensors, 18*(10), 3408. https://doi.org/10.3390/s18103408

Afzaal, H., Farooque, A., Abbas, F., Acharya, B., & Esau, T. (2020). Computation of evapotranspiration with artificial intelligence for precision water resource management. *Applied Sciences, 10*(5). https://doi.org/10.3390/app10051621

Agrosmart. (2021). Retrieved 11 1, 2021, from Agrosmart - Cultivo Inteligente: https://agrosmart.com.br/eng/.

Ahmed, A., Deo, R., Raj, N., Ghahramani, A., Feng, Q., Yin, Z., & Yang, L. (2021). Deep learning forecasts of soil moisture: Convolutional neural network and gated recurrent unit models coupled with satellite-derived MODIS, observations and synoptic-scale climate index data. *Remote Sensing, 4*. doi:10.3390/rs13040554.

Allen, R. G., Pereira, L. S., Raes, D., & Smith, M. (1998). *Crop evapotranspiration - Guidelines for computing crop water requirements*. Rome, Italy: Food and Agriculture Organization (FAO).

Alvino, A., & Marino, S. (2017). Remote sensing for irrigation of horticultural crops. (MDPI, Ed.). *Horticulturae, 3*(2). https://doi.org/10.3390/horticulturae3020040

Amarbayasgalan, T., Pham, V., Theera-Umpon, N., & Ryu, K. (2020). Unsupervised anomaly detection approach for time-Series in multi-domains using deep reconstruction error. *Symmetry, 12*(8). https://doi.org/10.3390/sym12081251

Anton, C., Matei, O., & Avram, A. (2019a). Collaborative data mining in agriculture for prediction of soil moisture and temperature. *Advances in Intelligent Systems and Computing, 984*, 141–151. https://doi.org/10.1007/978-3-030-19807-7_15

Anton, C., Matei, O., & Avram, A. (2019b). Use of multiple data sources in collaborative data mining. *Proceedings of the Computational Methods in Systems and Software, 1046*, 189–198. https://doi.org/10.1007/978-3-030-30329-7_18

Ardagna, D., Cappiello, C., Samá, W., & Vitali, M. (2018). Context-aware data quality assessment for big data. *Future Generation Computer Systems, 89*, 548–562. https://doi.org/10.1016/j.future.2018.07.014

Avram, A., Matei, O., Pintea, C.-M., & Pop, P. (2020). Context quality impact in context-aware data mining for predicting soil moisture. *Cybernetics and Systems, 51*(7), 668–684. https://doi.org/10.1080/01969722.2020.1798642

Avram, A., Matei, O., Pintea, C.-M., Pop, P., & Anton, C. (2019). Context-aware data mining vs classical data mining: Case study on predicting soil moisture. In *International Workshop on Soft Computing Models in Industrial and Environmental Applications* (pp. 199–208). https://doi.org/10.1007/978-3-030-20055-8_19

Beck, H., Pan, M., Miralles, D., Reichle, R., Dorigo, W., Hahn, S., … Wood, E. (2021). Evaluation of 18 satellite-and model-based soil moisture products using in situ measurements from 826 sensors. *Hydrology and Earth System Sciences, 25*, 17–40. https://doi.org/10.5194/hess-25-17-2021

Bertossi, L., & Geerts, F. (2020). Data quality and explainable AI. *Journal of Data and Information Quality (JDIQ), 12*(2), 1–9. https://doi.org/10.1145/3386687

Borisov, V., Leemann, T., SeBler, K., Haug, J., Pawelczyk, M., & Kasneci, G. (2022). Deep neural networks and tabular data: A survey. *arXiv:2110.01889v2*.

Cai, X., Hejazi, M., & Wang, D. (2011). Value of probabilistic weather forecasts: Assessment by real-time optimization of irrigation scheduling. *Journal of Water Resources Planning and Management, 137*(5). https://doi.org/10.1061/(ASCE)WR.1943-5452.0000126

Cai, S., Li, L., Li, S., Sun, R., & Yuan, G. (2020). An efficient approach for outlier detection from uncertain data streams based on maximal frequent patterns. *Expert Systems with Applications*, 160. https://doi.org/10.1016/j.eswa.2020.113646

Campo, L., Ledezma, A., & Corrales, J. (2020). Optimization of coverage mission for lightweight unmanned aerial vehicles applied in crop data acquisition. *Expert Systems with Applications, 149*. doi:10.1016/j.eswa.2020.113227.

Cao, W., Wang, D., Li, J., Zhou, H., Li, L., & Li, Y. (2018). BRITS: Bidirectional recurrent imputation for time series. *arXiv preprint arXiv:1805.10572*.

Cao, J., Tan, J., Cui, Y., & Luo, Y. (2019). March). Irrigation scheduling of paddy rice using short-term weather forecast data. *Agricultural Water Management, 223*(1), 714–723. https://doi.org/10.1016/j.agwat.2018.10.046

Cao, D., Wang, Y., Duan, J., Zhang, C., Zhu, X., Huang, C., … Zhang, Q. (2020). Spectral temporal graph neural network for multivariate time-series forecasting. *NeurIPS*. arXiv:2103.07719.

Carlson, T. N., & Ripley, D. A. (1997). December). On the relation between NDVI, fractional vegetation cover, and leaf area index. *Remote Sensing of Environment, 62*(3), 241–252. https://doi.org/10.1016/S0034-4257(97)00104-1

Chandola, V., Banerjee, A., & Kumar, V. (2009). Anomaly detection: A survey. *ACM Computing Surveys, 41*(3), 1–58. https://doi.org/10.1145/1541880.1541882

Chandrashekar, G., & Sahin, F. (2014). January). A survey on feature selection methods. *Computers & Electrical Engineering, 40*, 16–28.

Cui, D., Liang, S., & Wang, D. (2021). Observed and projected changes in global climate zones based on Köppen climate classification. *Wiley interdisciplinary reviews: Climate Change, 12*(5). doi:10.1002/wcc.701.

Domínguez-Niño, J., Oliver-Manera, J., Girona, J., & Casadesús, J. (2020). Differential irrigation scheduling by an automated algorithm of water balance tuned by capacitance-type soil moisture sensors. *Agricultural Water Management, 228*. https://doi.org/10.1016/j.agwat.2019.105880

Dubois, A., Teytaud, F., & Verel, S. (2021). Short term soil moisture forecasts for potato crop farming: A machine learning approach. *Computers and Electronics in Agriculture, 180*. doi:10.1016/j.compag.2020.105902.

Dubreil, V., Fante, K., Planchon, O., & Neto, J. (2018). Os tipos de climas anuais no Brasil : Uma aplicação da classificação de Köppen de 1961 a 2015. *Confins - Revista Franco-Brasileira de Geografia, 37*. https://doi.org/10.4000/confins.15738

Erhan, L., Ndubuaku, M., Di Mauro, M., Song, W., Chen, M., Fortino, G., … Liotta, A. (2021). March). Smart anomaly detection in sensor systems: A multi-perspective review. *Information Fusion, 67*, 64–79. https://doi.org/10.1016/j.inffus.2020.10.001

FAO. (2021a). *AquaCrop*. Retrieved 11 1, 2021, from https://www.fao.org/aquacrop.

FAO. (2021b). *Food and Agriculture Organization for the United Nations*. Retrieved from http://fao.org/home/en/.

Farthing, M., & Ogden, F. (2017). Numerical solution of Richards' equation: A review of advances and challenges. *Soil Science Society of America*, 1257–1269. https://doi.org/10.2136/sssaj2017.02.0058

Freedman, D. (2009). *Statistical models: Theory and practice*. Cambridge University Press.

García, I., Montesinos, P., Poyato, E., & Díaz, J. (2016). Energy cost optimization in pressurized irrigation networks. *Irrigation Science, 34*, 1–13. https://doi.org/10.1007/s00271-015-0475-3

García, L., Parra, L., Jimenez, J., Lloret, J., & Lorenz, P. (2020). IoT-Based Smart Irrigation Systems: An Overview on the Recent Trends on Sensors and IoT Systems for Irrigation in Precision Agriculture. *Sensors, 20*(4), 1042. https://doi.org/10.3390/s20041042

Géron, A. (2019). *Hands-on machine learning with Scikit-Learn, Keras, and Tensorflow: Concepts, tools, and techniques to build intelligent systems*. O'Reilly.

Goodfellow, I., Bengio, Y., & Courville, A. (2015). *Deep learning*. Cambridge - Massachussetts, London - England: The MIT Press.

Grafton, R., Williams, J., Perry, C., Molle, F., Ringler, C., Steduto, P., … Allen, R. (2018). The paradox of irrigation efficiency. *Science, 361*(6404), 748–750.

Gu, Z., Qi, Z., Ma, L., Gui, D., Xu, J., Fang, Q., … Feng, G. (2017). Development of an irrigation scheduling software based on model predicted crop water stress. *Computers and Electronics in Agriculture, 143*, 208–221. https://doi.org/10.1016/j.compag.2017.10.023

Gumière, S., Camporese, M., Botto, A., Lafond, J., Paniconi, C., Gallichand, J., & Rousseau, A. (2020). Machine learning vs. physics-based modeling for real-time irrigation management. *Water and Hydrocomplexity (Frontiers in Water)*. https://doi.org/10.3389/frwa.2020.00008

Hariri, R., Fredericks, E., & Bowers, K. (2019). Uncertainty in big data analytics: Survey, opportunities, and challenges. *Journal of Big Data, 6*(44). https://doi.org/10.1186/s40537-019-0206-3

Jang, J. (1993). ANFIS: Adaptive-network-based fuzzy inference system. *IEEE Transactions on Systems, Man, and Cybernetics, 23*(3), 665–685. https://doi.org/10.1109/21.256541

Jarman, M., & Dimmock, J. (2018). Satellites for Agriculture. AHDB / Catapult Satellite Applications.

Jensen, M., & Allen, R. (2016). Evaporation, evapotranspiration, and irrigation water requirements. *Task Committee on Revision of Manual, 70*. https://doi.org/10.1061/9780784414057

Junior, F., & Kamienski, C. (2021). March). A survey on trustworthiness for the internet of things. *IEEE Access, 9*, 42493–42514. https://doi.org/10.1109/ACCESS.2021.3066457

Kamienski, C., Soininen, J., Taumberger, M., Dantas, R., Toscano, A., Cinotti, T., … Torre Neto, A. (2019). Smart water management platform: IoT-based precision irrigation for agriculture. *Sensors, 19*(2), 276. https://doi.org/10.3390/s19020276

Karandish, F., & Šimůnek, J. (2016). A comparison of numerical and machine-learning modeling of soil water content with limited input data. *Journal of Hydrology, 543*(B), 892–909. https://doi.org/10.1016/j.jhydrol.2016.11.007

Kashyap, B., & Kumar, R. (2021). Sensing methodologies in agriculture for soil moisture and nutrient monitoring. *IEEE Access, 9*, 14095–14121. https://doi.org/10.1109/ACCESS.2021.3052478

Ke, G., Meng, Q., Finley, T., Wang, T., Chen, W., Ma, W., … Liu, T.-Y. (2017). LightGBM: A highly efficient gradient boosting decision tree. *Advances in Neural Information Processing Systems (NIPS), 30*.

Keras. (2021). *Keras*. Retrieved from Keras: https://keras.io/.

Kubat, M. (2018). *An introduction to machine learning*. Springer.

Linker, R., Ioslovich, I., Sylaios, G., Plauborg, F., & Battilani, A. (2016). Optimal model-based deficit irrigation scheduling using AquaCrop: A simulation study with cotton, potato and tomato. *Agricultural Water Management, 163*, 236–243. https://doi.org/10.1016/j.agwat.2015.09.011

Liu, Y., Gong, C., Yang, L., & Chen, Y. (2020). DSTP-RNN: A dual-stage two-phase attention-based recurrent neural network for long-term and multivariate time series prediction. *Expert Systems with Applications, 143*. doi:10.1016/j.eswa.2019.113082.

Liu, J., Li, Y., Huang, G., Zhuang, X., & Fu, H. (2017). Assessment of uncertainty effects on crop planning and irrigation water supply using a Monte Carlo simulation based dual-interval stochastic programming method. *Journal of Cleaner Production, 149*, 945–967. https://doi.org/10.1016/j.jclepro.2017.02.100

Luo, Y., Cai, X., Zhang, Y., Xu, J., & Yuan, X. (2018). Multivariate time series imputation with generative adversarial networks. *32nd Conference on Neural Information Processing Systems (NeurIPS)*. Montréal, Canada.

Luo, Y., Zhang, Y., Cai, X., & Yua, X. (2019). E2GAN: End-to-end generative adversarial network for multivariate time series imputation. *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence (IJCAI)*.

Makridakis, S., Spiliotis, E., & Assimakopoulos, V. (2020). The M4 Competition: 100,000 time series and 61 forecasting methods. *International Journal of Forecasting, 36*(1), 54–74. https://doi.org/10.1016/j.ijforecast.2019.04.014

Martins, F., Gonzaga, G., Santos, D., & Reboita, M. (2018). Classificação climática Köppen e de Thornthwaite para Minas Gerais: Cenário atual e projeções futuras. *Revista Brasileira de Climatologia*, pp. 129-156.

Matei, O., Rusu, T., Bozga, A., & Pop-Sitar, P. (2017). Context-aware data mining: Embedding external data sources in a machine learning process. *International Conference on Hybrid Artificial Intelligence Systems, 10334*, 415–426. https://doi.org/10.1007/978-3-319-59650-1_35

McElreath, R. (2020). *Statistical rethinking: A bayesian course with examples in R and Stan*. CRC Press.

Microsoft. (2021a). *LightGBM - Light gradient boosting machine*. Retrieved 11 1, 2021, from https://github.com/microsoft/LightGBM.

Microsoft. (2021b). *Spectral temporal graph neural network for multivariate time-series forecasting*. Retrieved 11 1, 2021, from https://github.com/microsoft/StemGNN.

Miller, P., Lanier, W., & Brandt, S. (2001). *Using growing degree days to predict plant stages*. Agriculture and Natural Resources.

Mohanty, B., Cosh, M., Lakshmi, V., & Montzka, C. (2017). Soil moisture remote sensing: State-of-the-science. *Vadose Zone Journal, 16*(1), 1–9. https://doi.org/10.2136/vzj2016.10.0105

Moon, A., Kim, J., Zhang, J., & Son, S. (2018). November). Evaluating fidelity of lossy compression on spatiotemporal data from an IoT enabled smart farm. *Computers and Electronics in Agriculture, 154*, 304–313. https://doi.org/10.1016/j.compag.2018.08.045

Oca, A., & Flores, G. (2021, 11 15). The AgriQ: A low-cost unmanned aerial system for precision agriculture. *Expert Systems with Applications, 182*. doi:10.1016/j.eswa.2021.115163.

Oreshkin, B., Carpov, D., Chapados, N., & Bengio, Y. (2019). N-BEATS: Neural basis expansion analysis for interpretable time series forecasting. *arXiv:1905.10437v4*.

Pattathal V, A., & Karnieli, A. (2022). Deep feature learning and latent space encoding for crop phenology analysis. *Expert Systems with Applications, 187*. doi:10.1016/j.eswa.2021.115929.

Pelikan, M., Goldberg, D., & Cantú-Paz, E. (1999). BOA: The Bayesian optimization algorithm. *Proceedings of the Genetic and Evolutionary Computation Conference*.

Pereira, L., Paredes, P., & Jovanovic, N. (2020). Soil water balance models for determining crop water and irrigation requirements and irrigation scheduling focusing on the FAO56 method and the dual Kc approach. *Agricultural Water Management, 241*(1). doi:10.1016/j.agwat.2020.106357.

Raubitzek, S., & Neubauer, T. (2021). A fractal interpolation approach to improve neural network predictions for difficult time series data. *Expert Systems with Applications, 169*. doi:10.1016/j.eswa.2020.114474.

Rawls, W., Gish, T., & Brakensiek, D. (1991). Estimating soil water retention from soil physical properties and characteristics. *Advances in Soil Science, 16*, 213–234. https://doi.org/10.1007/978-1-4612-3144-8_5

Reddy, P. (2017, 11). Types of irrigation and historical development - A comprehensive compilation. *Journal of Indian Geophysical Union, 21*(6), pp. 535-542.

Roberts, D., Bahn, V., Ciuti, S., Boyce, M., Elith, J., Guillera-Arroita, G., … Dormann, C. (2017). Cross-validation strategies for data with temporal, spatial, hierarchical, or phylogenetic structure. *Ecography, 40*, 913–929. https://doi.org/10.1111/ecog.02881

Rockefeller, B. (2019). *Technical Analysis for Dummies*. For Dummies.

Rohli, V., Joyner, T., Reynolds, S., & Ballinger, T. (2015, 3). Overlap of global Köppen–Geiger climates, biomes, and soil orders. *Physical Geography, 36*(2). doi:10.1080/02723646.2015.1016384.

Scikit-Learn. (2021a). *Metrics and scoring: quantifying the quality of predictions*. Retrieved 11 1, 2021, from https://scikit-learn.org/stable/modules/model_evaluation.html.

Scikit-Learn. (2021b, 11 1). *sklearn.metrics.mean_squared_error*. Retrieved from Scikit-Learn: https://scikit-learn.org/stable/modules/generated/sklearn.metrics.mean_squared_error.html.

Scikit-Learn. (2021c). *sklearn.metrics.mean_absolute_error*. Retrieved 11 1, 2021, from https://scikit-learn.org/stable/modules/generated/sklearn.metrics.mean_absolute_error.html#sklearn.metrics.mean_absolute_error.

Scikit-Learn. (2021d). *sklearn.metrics.r2_score*. Retrieved 11 1, 2021, from https://scikit-learn.org/stable/modules/generated/sklearn.metrics.r2_score.html?highlight=r2#sklearn.metrics.r2_score.

Scikit-Learn. (2021e). *sklearn.metrics.mean_absolute_percentage_error*. Retrieved 11 1, 2021, from https://scikit-learn.org/stable/modules/generated/sklearn.metrics.mean_absolute_percentage_error.html#sklearn.metrics.mean_absolute_percentage_error.

Scikit-Learn. (2021f). *scikit-learn*. Retrieved from scikit-learn: https://scikit-learn.org/stable/.

scikit-optimize. (2022). *skopt.gp_minimize*. Retrieved from scikit-optimize: https://scikit-optimize.github.io/stable/modules/generated/skopt.gp_minimize.html#skopt.gp_minimize.

Shahdany, S., Maestre, J., & van Overloop, P. (2015). Equitable water distribution in main irrigation canals with constrained water supply. *Water Resources Management, 29*, 3315–3328. https://doi.org/10.1007/s11269-015-1000-4

Shang, C., Chen, W.-H., Stroock, A., & You, F. (2020). July). Robust model predictive control of irrigation systems with active uncertainty learning and data analytics. *IEEE Transactions on Control Systems Technology, 28*(4), 1493–1504. https://doi.org/10.1109/TCST.2019.2916753

Šimůnek, J., van Genuchten, M., & Šejna, M. (2008). Development and applications of the HYDRUS and STANMOD software packages, and related codes. *Vadose Zone Journal, 7*(2), 587–600. https://doi.org/10.2136/VZJ2007.0077

Šimůnek, J., van Genuchten, M., & Šejna, M. (2016). Recent developments and applications of the HYDRUS computer software packages. *Vadose Zone Journal, 15*(7), 25. https://doi.org/10.2136/vzj2016.04.0033

Singh, B., Deznabi, I., Narasimhan, B., Kucharski, B., Uppaal, R., Josyula, A., & Fiterau, M. (2019). Multi-resolution networks for flexible irregular time series modeling (Multi-FIT). *arXiv:1905.00125*.

Sishodia, R., Ray, R., & Singh, S. (2020). Applications of remote sensing in precision agriculture: A review. *Remote Sensing, 12*(19), 3136. https://doi.org/10.3390/rs12193136

Srikanth, T., Branch, P., Jin, J., & Jugdutt, S. (2020). A comprehensive survey of anomaly detection techniques for high dimensional big data. *Journal of Big Data, 7*(42). https://doi.org/10.1186/s40537-020-00320-x

Sun, Z., Di, L., Fang, H., Guo, L., Tan, X., Jiang, L., & Shen, Z. (2021). Agro-geoinformatics Data Sources and Sourcing. *Agro-geoinformatics*, pp. 41-66. doi:10.1007/978-3-030-66387-2_4.

Togneri, R., Camponogara, G., & Kamienski, C. (2019). Foundations of data quality assurance for IoT-based smart applications. In *IEEE Latin-American Conference on Communications (LATINCOM)*. Salvador, Brazil: IEEE. https://doi.org/10.1109/LATINCOM48065.2019.8937930.

Topp, G., Parkin, G., & Ferré, T. (2008). Soil water content. In M. Carter, & E. Gregorich (Eds.), *Soil sampling and methods of analysis*. Canadian Society of Soil Science.

Torres, A., da Rocha, A., da Silva, T., de Souza, J., & Gondim, R. (2020). April). Multilevel data fusion for the internet of things in smart agriculture. *Computers and Electronics in Agriculture, 171*. https://doi.org/10.1016/j.compag.2020.105309

Torres, M., Colominas, M., Schlotthauer, G., & Flandrin, P. (2011). A complete ensemble empirical mode decomposition with adaptive noise. *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. Prague, Czech Republic. doi:10.1109/ICASSP.2011.5947265.

Torres, A., Walker, W., & McKee, M. (2011). February). Forecasting daily potential evapotranspiration using machine learning and limited climatic data. *Agricultural Water Management, 98*(4), 553–562. https://doi.org/10.1016/j.agwat.2010.10.012

Toscano, A., Stanghellini, C., Bittelli, M., Castaldi, P., Soininen, J.-P., Torre Neto, A., . . . Ricchi, T. (2019). *Deliverable D3.1 (water need estimation) of SWAMP (smart water management platform)*.

UCAR. (2021). *The weather research and forecasting model*. Retrieved 11 1, 2021, from https://www.mmm.ucar.edu/weather-research-and-forecasting-model.

Van Genuchten, M. (1980). A closed-form equation for predicting the hydraulic conductivity of unsaturated soils. *Soil Science Society of America Journal, 44*(5), 892–898. https://doi.org/10.2136/sssaj1980.03615995004400050002x

Wang, Y., Yuan, Z., Liu, H., Xing, Z., Ji, Y., Li, H., … Mo, C. (2022). A new scheme for probabilistic forecasting with an ensemble model based on CEEMDAN and AM-MCMC and its application in precipitation forecasting. *Expert Systems with Applications, 187*. doi:10.1016/j.eswa.2021.115872.

Wrege, M., Steinmetz, S., Júnior, C., & Almeida, I. (2012). *Atlas climático da região sul do Brasil - Estados do Paraná, Santa Catarina e Rio Grande do Sul*. Brasília: Embrapa.

Xu, B., Shao, D., Tan, X., Yang, X., Gu, W., & Li, H. (2017). Evaluation of soil water percolation under different irrigation practices, antecedent moisture and groundwater depths in paddy fields. *Agricultural Water Management, 192*, 149–158. https://doi.org/10.1016/j.agwat.2017.06.002

Yildiz, C., Heinonen, M., & Lähdesmäki, H. (2019). ODE2VAE: Deep generative second order ODEs with Bayesian neural networks. *33rd Conference on Neural Information Processing Systems (NeurIPS)*. Vancouver, Canada.

Yu, J., Zhang, X., Xu, L., Dong, J., & Zhangzhong, L. (2021). A hybrid CNN-GRU model for predicting soil moisture in maize root zone. *Agricultural Water Management, 245*. https://doi.org/10.1016/j.agwat.2020.106649

Zeitoun, R., Vandergeest, M., Vasava, H., Machado, P., Jordan, S., Parkin, G., … Biswas, A. (2021). In-situ estimation of soil water retention curve in silt loam and loamy sand soils at different soil depths. *Sensors, 21*(2), 447. https://doi.org/10.3390/s21020447