A Virtual Soil Moisture Sensor for Smart Farming Using Deep Learning

Gabriele Patrizi, *Member, IEEE*, Alessandro Bartolini, *Graduate Student Member, IEEE*, Lorenzo Ciani, *Senior Member, IEEE*, Vincenzo Gallo, *Graduate Student Member, IEEE*, Paolo Sommella, *Member, IEEE*, and Marco Carratù, *Member, IEEE*

Abstract—**Precision farming technologies refer to a set of cutting-edge tools and strategies implemented to optimize the management of the plantation. Smart meter devices, Internet of Things (IoT) technologies, and wireless sensor networks (WSNs) are only a few examples of the innovative systems increasingly employed from an Agriculture 4.0 point of view. Recent literature has paid close attention to the role of artificial intelligence (AI) and deep learning (DL) algorithms in helping farmers and improving soil productivity. In this regard, this article presents the design of a WSN based on low-cost, low-power photovoltaic (PV)-supplied sensor nodes able to acquire data regarding environmental conditions and soil parameters. Among all the implemented sensors, the most critical is the soil moisture sensors because of many issues related to cost, installation, reliability, and calibration. Thus, this article proposes a DL approach based on long short-term memory (LSTM) networks to provide a virtual soil moisture sensor using only the data acquired by the other transducer installed on the node. Performance estimation of the virtual sensors and an in-depth comparison with other learning-based approaches have been presented in this article to validate the effectiveness of the proposed soft sensing approach.**

Index Terms—**Agricultural engineering, artificial intelligence (AI), long short-term memory (LSTM), sensor fusion, smart devices, soft sensors, wireless sensor networks (WSNs).**

I. Introduction

TECHNOLOGICAL evolutions over the last decade are making predictive modeling processes in multiple fields ever easier. In particular, the massive diffusion of artificial neural networks (ANNs) and, more generally, deep learning (DL) algorithms make computers more autonomous in learning processes for classifications, regressions, and predictions. This evolution is of fundamental importance in the Industry 4.0 paradigm, where these innovative analysis tools are integrated within the context of smart metering based on Internet of Things (IoT) architectures. Thanks to the latest evolutions of artificial intelligence (AI), it is possible to analyze the huge amount of data produced by this growing number of intelligent sensor nodes in order to extract useful information for the optimization of industrial processes [1], [2].

This optimization concerns the industrial world and any other sector where reducing costs is necessary. Among these fields, the interest in AI-based optimization of IoT architectures is rapidly spreading in the agricultural industry [3], [4]. As a matter of fact, precision farming (also known as smart farming) technologies involve the use of smart meters, wireless sensor networks (WSNs), unmanned aerial vehicles (UAVs), smart camera nodes, and many other IoT technologies in order to fulfill the following [5]–[7].

1) Decrease the use of pesticides allowing a more sustainable and eco-friendly production.
2) Optimize the irrigation reducing water waste.
3) Enhance the overall resources management of the farm.
4) Diagnose and prevent plant diseases allowing a significant improvement in the harvest.
5) Reduce farming costs.

A. Problem Background

Smart farming is crucial in the fight against climate change, as the waste of natural resources contributes substantially to worsening the situation. In this regard, the world of agriculture is pushing toward improving monitoring efficiency by employing DL algorithms as decision support in agricultural activities [8].

More in detail, the improvement in decision making, with consequent optimization of the use of natural and economic resources, has been studied for multiple activities related to this area. For example, environmental factors capable of affecting the crop yield and the consequent irrigation scheduling were analyzed in [9] using DL algorithms. Furthermore, the study of factors that may influence irrigation scheduling was analyzed with convolutional neural networks (CNNs) and long short-term memory (LSTM) networks in [10]. Similarly, a DL approach for smart irrigation systems is proposed in [11] using an LSTM-based neural network.

Another example of the use of DL algorithms is given by Moon *et al.* [12], where a prediction of CO_2 concentration in greenhouses was made thanks to an LSTM network trained with different environmental data, including temperature, relative humidity (%RH), atmospheric pressure, CO_2 concentration, soil parameters (temperature, humidity, and electrical conductivity), and parameters outside the greenhouse (temperature, humidity, and wind).

Disease detection is another central point of study in this context that could benefit significantly from the introduction

Manuscript received 14 June 2022; revised 12 July 2022; accepted 23 July 2022. Date of publication 4 August 2022; date of current version 12 August 2022. The Associate Editor coordinating the review process was Huang-Chen Lee. *(Corresponding author: Lorenzo Ciani.)*

Gabriele Patrizi, Alessandro Bartolini, and Lorenzo Ciani are with the Department of Information Engineering, University of Florence, 50139 Florence, Italy (e-mail: gabriele.patrizi@unifi.it; a.bartolini@unifi.it; lorenzo.ciani@unifi.it).

Vincenzo Gallo, Paolo Sommella, and Marco Carratù are with the Department of Industrial Engineering, University of Salerno, 84084 Fisciano, Italy (e-mail: vgallo@unisa.it; psommella@unisa.it; mcarratu@unisa.it).

Digital Object Identifier 10.1109/TIM.2022.3196446

of AI. For instance, a shallow visual geometric group (VGG) based on several convolutional and pooling layers has been introduced in [13] to identify plant diseases. The effectiveness of the network has been tested using images of potato, corn, and tomato fields. Other approaches for plant disease identification and classification include the use of adapted deep residual neural network-based algorithms [14], LSTM [15], transfer learning [16], and embedded system enriched with a recurrent neural network (RNN) [17]. For a more detailed discussion about this topic, see [18] and [19].

Soft sensors, also known as virtual sensors, have also found great engagement in this context. This term refers to a software node where numerous measurement tasks are performed and processed together. The spread of ANNs has made possible the development of increasingly advanced soft sensors capable of identifying correlations between measures that could not be predicted in the past. For example, in [20], a soft sensor for gasoline engine exhaust emission estimation has been devised using ANNs. Soft sensors are crucial in environments where it is impossible for logistical reasons, such as installing physical sensors [21], [22]. Their use is also helpful in areas where the failure of one of the sensors can cause severe damage, both functionally and economically. This aspect is mainly described in [23], where soft sensors are implemented for instrument fault accommodation, and in [24], where soft sensors were used for timber bundle volume measurement in the Swedish forest industry.

In this context, the applicability of the soft sensor concept to the precision agriculture field employing DL algorithms was evaluated. In particular, a study has been carried out on the cost of all the sensors necessary for the correct irrigation programming of the fields to be cultivated. The results of the analysis emphasized that the soil moisture measurement sensor is one of the most important but also one of the most expensive and most critical sensor in a weather station for agricultural purposes, as widely described in [11] and [25]-[27].

B. Literature Review

So far, soil moisture is estimated using WSNs and satellite imagery, but these are expensive and have low accuracy [28]. To overcome these limitations, the researchers have tried to make the most of all the sensors present above and below ground to reduce costs and increase the accuracy of the analysis and prediction of the parameters. Specifically, Liu *et al.* [29] tried to predict soil moisture by employing standard environmental sensors comprising ambient temperature and humidity sensors and the soil moisture sensor. The authors then implemented a smart sensor node connected with the cloud platform via the Message Queue Telemetry Transport (MQTT) protocol. However, the approach's limitations can be traced to two factors: the first relates to the economic factor since, as mentioned earlier, soil moisture sensors are not cheap; moreover, the proposed method requires multiple sensors to improve the accuracy of measurements over a large area of land. Another limitation is the possible failures that may have the sensors, more likely due to their greater quantity, combined with possible measurement errors due to hostile environmental conditions and improper installation by the operators.

Another approach presented in [30] uses a support vector machine (SVM) to predict soil moisture based on digital photography. However, the cost and size of the required hardware make the approach based on photography not always affordable. Another approach based on soil picture is presented in [33]. Different machine learning methods for soil moisture prediction are investigated in [31]. This article presents a remarkable review of machine learning approaches; however, it misses considering RNN and their improvements, such as LSTM. Another comparison between machine learning algorithms for soil moisture prediction has been proposed in [32], considering multiple linear regression, support vector regression, and RNN. The major limitation of the work is the limited time forecast (up to 7 days). Other available approaches include the use of graph neural networks [34], intelligent multioutput regression model [35], and DL regression network [36].

Therefore, to comply with the soft sensor concept, it was necessary to supplement the collected data with an estimation and prediction algorithm from exogenous variables related to the variable. However, in such cases, correlations between environmental variables are often weak and insufficient for applying classical estimation and prediction algorithms such as Kalman filtering and Fuzzy logic.

C. Major Contributions

For the reasons mentioned above, it was decided to use the black box approach concept typical of DL algorithms, which are used in almost every field today. In particular, it was decided to leverage the capabilities of LSTM networks in conjunction with the soft sensor concept to remove the soil moisture sensor by predicting the value mentioned above from other environmental measurements taken at the site. Furthermore, this method is also characterized by the ability to predict not only the present value of soil moisture but also more distant time steps due to the structure of the LSTM network.

The choice of LSTM networks is based not only on the excellent performance in regression and prediction by the network itself but also on the scalability and flexibility capabilities of the technique. In fact, the neural network can be retrained to accommodate a change in geographic sensor placement; it is also possible to change the architecture slightly to vary the number of inputs while keeping the final soft sensor output unchanged. In this way, it is possible to integrate new sensors without overcomplicating the architecture, which can improve performance during the regression process of the network.

The major contributions of this article are as follows.

1) The introduction of a DL technique within a traditional monitoring device for smart pharming technologies. This allows to propose an innovative AI-based IoT device for precision agriculture that is currently not available neither in the literature nor in the market.
2) The implementation and characterization of a virtual soil moisture sensor using the LSTM network, which to the author's knowledge, has never been developed in the literature.

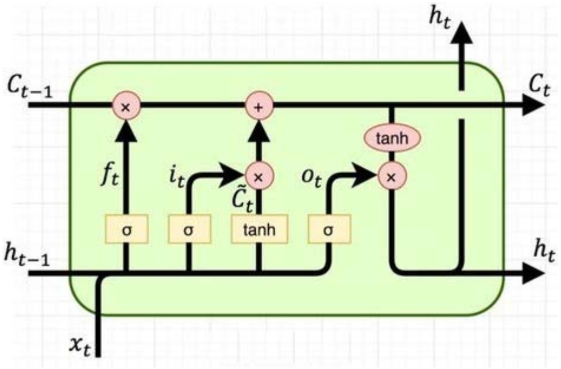

Fig. 1. Schematic representation of the LSTM cell architecture [37].

3) The optimization of both LSTM network architecture and hyperparameters with respect to the problem under study in order to improve the accuracy of prediction.
4) A detailed comparison between LSTM, fully RNN (FRNN), nonlinear autoregressive network with exogenous inputs (NARX), and multivariate regression tree for virtual soil moisture sensor, which to the author's knowledge, is missing in recent literature.

D. Manuscript Structure

This article will introduce the concept of LSTM units and how they integrate into more complex architectures in Section II. Then the experimental setup will be described from a hardware point of view in Section III. In Section IV, the data analysis and the neural network training will be described, along with all the baseline models available in state of the art, chosen for performance comparison. Section V will compare the results with a traditional nonlinear regression method, a NARX networks, and a FRNN. Finally, in Section VI, the conclusions will be presented.

II. DL FOR SOFT SENSING

A. LSTM Networks

LSTM networks are special neural networks composed of individual LSTM units. These units result from an evolution of the RNNs, which are neural networks having a state, a feature that allows them to learn characteristics of long temporal sequences, to make classifications or predictions on future predictions time steps.

The need to use LSTMs instead of RNNs arises from the "vanishing" gradient problem, i.e., the possibility that the gradient may vanish or explode during the back-propagation operation, the real pillar of the training of ANNs. An example of an LSTM network structure can be seen in Fig. 1. In particular, the signals shown in the figure are described by the set of following equations:

$$i_t = \sigma\left(x_t U^i + h_{t-1} W^i\right) \tag{1}$$

$$f_t = \sigma\left(x_t U^f + h_{t-1} W^f\right) \tag{2}$$

$$o_t = \sigma\left(x_t U^o + h_{t-1} W^o\right) \tag{3}$$

$$\tilde{C}_t = \tanh\left(x_t U^g + h_{t-1} W^g\right) \tag{4}$$

$$C_t = \sigma \times \left(f_t \times C_{t-1} + i_t \times \tilde{C}_t\right) \tag{5}$$

$$h_t = \sigma_t \times \sigma_h(c_t) \tag{6}$$

where σ is a sigmoid activation function, σ_t and σ_h are two hyperbolic tangent activation functions, f_t is the forget gate's activation vector, i_t is the input/update gate's activation vector, o_t is the output gate's activation vector, h_t is the hidden state vector, $\tilde{C}_t$ is the cell input activation vector, and C_t is the cell state vector. The different W stand for the weight matrices and bias vector parameters learned during the training.

The three gates regulate the flow of information through the cell since the number of states stored is arbitrarily defined during training.

LSTM networks can be used, depending on the choice of the output layer, either for the classification of temporal sequences or for the prediction of successive temporal instants.

In particular, in order to make a convincing prediction, the network must analyze several samples, defined as a warmup, that are not given as output by the network.

In addition to the reference time steps, these networks can have one input and one output (univariate LSTM) or multiple inputs and one output (multivariate LSTM). The latter approach was the appropriate one for the case study, as it involves an improvement in the quality of the prediction of the output variable due to the simultaneous analysis of multiple variables. Furthermore, this approach allows the identification of correlations between the input and output variables that are often difficult to identify with traditional methodologies and approaches.

The soft sensor can then be designed from a training having as input the measured variables and as output, the objective measurement of the soft sensor, to physically remove the sensor after the training and continue to estimate its measurement at the software level.

B. Dataset Construction

As in all DL approaches, it is necessary to construct the dataset respecting size, intraclass variability, and interclass variability criteria. In this case, it is necessary to ensure a sufficient number of samples for the warmup of the network. Furthermore, the dataset must be made up of the input data and the corresponding time step, which is essential for predicting future outputs of the time series analysis.

The neural network then accepts input in a format (batch, sequence length, input features), provided following a structure visible in Fig. 2.

Another fundamental aspect regarding the construction of the dataset is its normalization. This is crucial because a dataset normalized according to a Gaussian simplifies the optimizer's work considerably, thus increasing the performance of the prediction. The equation used for the normalization of the dataset is (7), where F_i stands for the input feature, or input data, NF for the Normalized Features, $\bar{F}$ for the mean of the features, and s_F for the sample standard deviation of the features

$$\mathrm{NF}_i = \frac{F_i - \bar{F}}{s_F}. \tag{7}$$

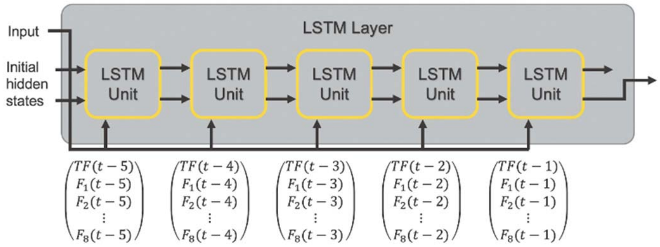
Input
Initial hidden states
LSTM Layer
LSTM Unit
LSTM Unit
LSTM Unit
LSTM Unit
LSTM Unit
$TF(t-5)$ $F_1(t-5)$ $F_2(t-5)$ ⋮ $F_8(t-5)$
$TF(t-4)$ $F_1(t-4)$ $F_2(t-4)$ ⋮ $F_8(t-4)$
$TF(t-3)$ $F_1(t-3)$ $F_2(t-3)$ ⋮ $F_8(t-3)$
$TF(t-2)$ $F_1(t-2)$ $F_2(t-2)$ ⋮ $F_8(t-2)$
$TF(t-1)$ $F_1(t-1)$ $F_2(t-1)$ ⋮ $F_8(t-1)$

Fig. 2. Example of LSTM input structure [38].

III. Experimental Setup

This section presents the characteristics of the proposed IoT infrastructure for smart farming technologies. The entire system is a WSN based on a self-organized and self-configured mesh topology to collect data from a distributed network of sensor nodes deployed on a wide field. The nodes are independent and self-powered units that collect different kinds of environmental data and transmit them to a central gateway (i.e., the root node or simply the access point). Furthermore, they also serve as a relay for other nodes to reach the central gateway using a multihop protocol and dynamic routing tables.

The proposed sensor node is a low-cost and low-power device composed of the following subunits.

1) The power supply unit uses the energy provided by a photovoltaic (PV) panel to recharge two lithium ion batteries through a customized dc–dc converter.
2) The acquisition and elaboration unit uses the ESP32 system-on-a-chip microcontroller to acquire data from analog and digital sensors and transmit them to the central gateway through IEEE 802.11 Wi-Fi protocol. Digital sensors are managed directly by the microcontroller, while analog sensors are acquired through adequate and customized conditioning circuits and two embedded eight-channel 12-bit successive approximation register (SAR) analog-to-digital converter (ADC).
3) An external 3 dBi antenna.
4) A sensors pack, including the following transducers.
   a) The AM2315 digital temperature and humidity sensor. The temperature measurement is characterized by a 0.1 °C and 16-bit resolution ranging from −40 °C to 125 °C. The typical accuracy is ±0.1 °C in the center of the range, while it can reach the maximum value of ±1 °C in the proximity of the range's bound. Instead, the relative humidity measurement has a typical resolution of 0.1 %RH and typical accuracy of ±2 %RH in the wide range [0, 99.9] %RH. This sensor is used to acquire ambient temperature (At) and relative humidity of the air (RHa).
   b) The DS18B20 programmable (9- to 12-bit resolution) digital temperature sensor is used to acquire the soil temperature (St). It is characterized by ±0.5 °C accuracy from −10 °C to 85 °C.
   c) The SM 100 soil moisture sensor, which is an analog transducer characterized by a 3% volumetric water content (VWC) accuracy and a 0.1% VWC resolution. This sensor is used to collect information about the relative humidity of the soil (VWC), which represents the target feature to be estimated using the proposed soft sensing algorithm.
   d) An analog solar radiation sensor used to collect the light radiation (Rad).
   e) An embedded thermistor with ±0.5 °C accuracy implemented to monitor the microcontroller's overheating.

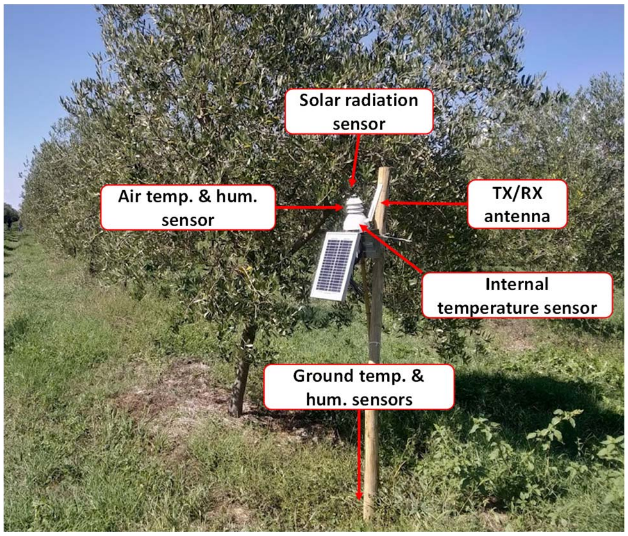

Fig. 3. Photograph of a sensor node installation on the olive grove.

The duty cycle of the proposed sensor node is almost 5% (i.e., the node works 1 min to acquire and send data over a period of 20 min) to minimize energy consumption, to avoid an oversampling of environmental data (note that meteorological parameters change slowly during the day), and to reduce component overheating and thus increase system reliability. See [37] and [38] for more information about the developed hardware.

The proposed network has been installed within an olive grove nearby Pisa, Tuscany, central Italy. A set of ten battery-powered sensor nodes have been deployed in different locations of the olive grove, and they have been used to acquire environmental and soil parameters for more than one year. An example of sensor node installation on the field is illustrated in Fig. 3, highlighting the location of the different implemented sensors with dedicated labels.

The proposed WSN for smart farming applications has been self-developed, taking into account the following principles.

1) Low-cost hardware able to ensure a good trade-off between metrological performances, computational complexity, and the price of the final product.
2) Fault-tolerant network architecture to ensure high-reliability requirements.
3) Easiness and quickness of installation on the field.
4) Low power consumption to maximize the battery life and minimize the need for maintenance operations due to loss of power.
5) Easiness of sensor calibration.

Considering all the principles mentioned above, it stands rapidly out of the numerous issues and challenges brought by the SM 100 soil moisture sensor. First of all, the high cost of the soil moisture sensor makes it extremely difficult to maintain a good tradeoff between the performance and price of the final product. As a matter of fact, half the hardware development cost of the entire sensor node (including all the sensors and the required electronics) is due only to the SM 100 soil moisture sensor. Furthermore, the soil moisture sensor also has a higher cost in terms of maintenance. Moreover, several studies on the prototypes highlighted that the soil moisture sensor has a higher probability of failure than all the other sensors installed in the proposed node, with a major impact on the overall system's reliability.

Moreover, the calibration process of the SM100 is extremely difficult, and the installation is quite challenging too. It must be deployed on the ground at a precise depth level, which must be exactly the same for all the sensors in the WSN to ensure consistent results. Considering all the limitations mentioned above, a DL approach for soft sensing has been implemented in this work. The work aims to substitute the soil moisture transducer with a virtual sensor achieved by utilizing an LSTM network. The data acquired by the other sensors installed in the station are used to train the network and predict the target feature (i.e., the VWC).

IV. Data Analysis and Network Training

A. Data Correlations

Following the data collection, organized in tables, an in-depth analysis was carried out to remove any errors and to understand the presence of any correlations capable of improving the ability to estimate the relative humidity of the soil. As indicated above, the data available concern the time of acquisition, i.e., the timestamp, the ambient temperature (At), the RHa, the soil temperature (St), the relative humidity of the soil (target feature, VWC), the light radiation (Rad), and other values concerning the operating status of the smart sensor. The first results concerning this analysis are shown in Fig. 4, where data concerning environmental and soil parameters have been examined. The matrix presents on the major diagonal unitary values, as they refer to the autocorrelation of the features. In contrast, in the other cells, values between -1 and 1 represent the correlation of covariances between the features reported, respectively, in the specific row and column. As can be seen from the correlation matrix, the features that present greater correlations with the others have been air temperature, relative air humidity, and soil temperature. On the contrary, the light radiation and the relative humidity of the soil had less strong correlations with the other features.

The main focus of this discussion was that soil moisture had stronger correlations with air temperature and soil temperature, while it has weaker correlations with relative air humidity and light radiation. Therefore, to improve the soil moisture prediction as much as possible, it was decided to include additional data regarding the amount of rainfall. Although no rainfall sensor was installed on the site, it was possible to obtain the above data related to the specific location where the sensors were installed, thanks to the hydrological and geological service of the Tuscany Region, Italy.

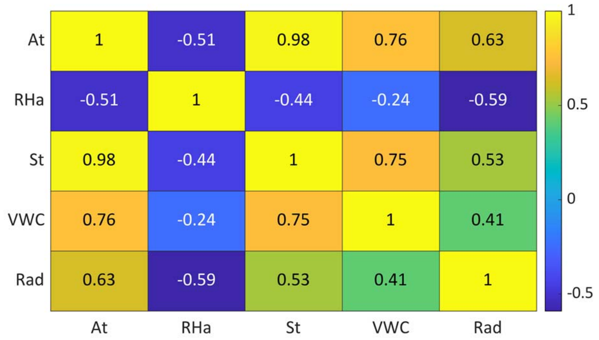

Fig. 4. Correlation heatmap between data acquired by the proposed sensor node.

TABLE I

Extract of the Complete Data Format Used in this Work

Date Time	*At (°C)*	*RHa (%)*	*St (°C)*	*VWC (%)*	*Rad (Lux)*	*Rain (cm/h)*
20/10/2020	*7.2*	*90.1*	*7.8*	*35.3*	*0.2*	*0.5*

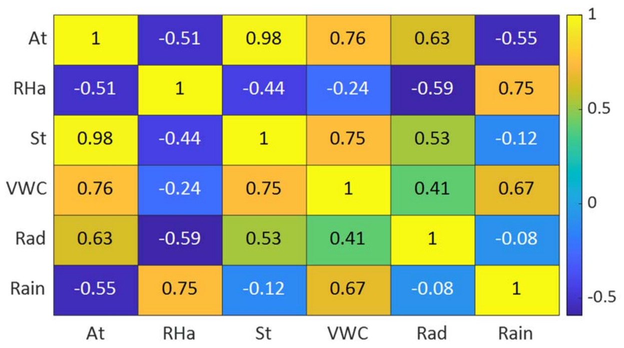

Fig. 5. Correlation heatmap including sensors installed on the prototype and rainfall data.

Following the data collection, it was necessary to synchronize them, sampled every 15 min, with those acquired by the setup described above. An example of the complete data format can be seen in Table I.

Finally, the confusion matrix was calculated similar to what was done before. The results are shown in Fig. 5. It can be seen from the correlation heatmap that rainfall data have a weak negative correlation with air temperature, a medium correlation with air humidity and soil moisture, and no correlation with soil temperature and light radiation. The evaluation of the significance of the input variables for the algorithms employed has been reported in [39] and states that the variable is significant if its correlation is greater than the following equation, for large N, numerosity of input data

$$\text{Absolute correlation} \geq \frac{2}{\sqrt{N}}. \tag{8}$$

B. Multivariate Regression Tree

As described above, it was not possible to perform a linear regression to predict soil moisture values, as there are no sufficiently strong correlations between the different features. Therefore, it was decided to use a regression method based on a specific machine learning algorithm. Thus, in the form of a multivariate regression tree, decision tree learning has been used as a term of comparison for the performances of the proposed DL algorithm.

This method is, in fact, used in data mining, machine learning, and statistical applications for both classification and decision-making applications and prediction applications [40], [41].

In particular, three specific profiles of this algorithm were used: fine tree, medium tree, and coarse tree.

The parameter used by the algorithm as a minimization objective is the entropy $E(S)$ (9)

$$E(S) = \sum_{i=1}^{N} \left(-p_i \log_2 p_i\right) \tag{9}$$

where S is the set of all instances in the dataset, N is the number of distinct class values, and p_i is the event probability.

In this case, it can measure the impurity of the sub splits of the regression tree, improving the distinction of the features and, therefore, the successive regressions. In addition to this parameter, the Gini impurity I_G (10) is also usually used in constructing the regression tree, which is also useful in estimating the purity of the tree's branches

$$I_G= 1-\sum_{j=1}^{N} \left(p_i^2\right) \tag{10}$$

where N is the number of distinct class values, and p_i is the event probability.

The model was generated on one dataset and then tested on three datasets in all cases. These results are reported in Section V.

C. NARX Model

Another tested methodology belonging to the field of neural networks is NARX. These networks are part of state of the art in time series classification and prediction. The model relates the current value of a time series to the past values of the same series and to the current and past values of the driving factor (the model).

The NARX model can be formalized as follows:

$$y_t = F(y_{t-1}, y_{t-2}, y_{t-3}, \ldots, u_t, u_{t-1}, u_{t-2}, u_{t-3}, \ldots,) + \varepsilon. \tag{11}$$

In (10), y_t is the variable of interest at time t, u_t is the externally determined variable at time t, and ε is the error.

The designed NARX network has been structured with ten hidden layers and one delay step.

D. Fully Recurrent Neural Network

FRNNs are deep neural networks capable of classifying and predicting temporal sequences characterized by the greatest simplicity of hyperparameters possible. The FRNN is a simple deep neural network where, however, every neuron, except those belonging to the input layer, has a feedback link from the output to the input of another neuron. This is also the case for multiple hidden layers and output layer neurons. This greatly simplifies the basic cell structure at the expense of architectural complexity [42].

TABLE II
ARCHITECTURE OF THE PROPOSED FRNN

Layer	*Type*	*Shape*
1	*Input*	*(Batch, 2000, 5)*
2	*Fully Recurrent Unit*	*(Batch,32)*
3	*Dense (Output)*	*(Batch,1)*

TABLE III
ARCHITECTURE OF THE PROPOSED LSTM NETWORK

Layer	*Type*	*Shape*
1	*Input*	*(Batch, 2000, 5)*
2	*LSTM*	*(Batch,32)*
3	*Dense (Output)*	*(Batch,1)*

The architecture of FRNN employed in the work is shown in Table II and includes 2000 warmup samples, 32 fully recurrent units, and one neuron for the output layer for the regression task.

E. LSTM Network

As described in Section II, the choice of DL algorithm fell on LSTM Networks. In particular, tests were carried out to evaluate the performance of the different hyperparameters, such as the length of the sequence for warming up the network, the sampling rate, the batch size, the number of epochs, the learning rate, the training, and validation split of the data.

The network architecture was then designed, as shown in Table III. The size of the input layer had to take into account the number of input variables, namely, St, RHa, Gt, Rad, and Rain. So, the shape of the input level is composed of the batch, the warmup steps (2000), and the number of input variables, while the LSTM level is composed of 32 units with no activation function. The output layer consists of only one neuron to comply with the regression task. The training was monitored with the mean square error loss parameter and using the Adam optimizer.

F. Network Training

In order to train the network, it was first necessary to fix all the hyper-parameters in order to reduce the mean squared error of the neural network output compared to the validation set. Although there is no general rule for defining the architecture of the neural network, the literature has reported that the number of hidden layers should be only one, or at least no more than two, to keep accuracy high and reduce possible overfitting [43].

The number of LSTM cells of the hidden layer, on the other hand, is highly dependent on the level of complexity of the input signal.

For this reason, it should be as few as possible in order to avoid overfitting problems. Therefore, the operation was carried out empirically, verifying each time the trend of the

TABLE IV
BEST PARAMETERS OF THE PROPOSED LSTM NETWORK

Hyperparameter	*Value*
Learning Rate	*1e-4*
Batch Size	*128*
Epochs	*40*
Optimizer	*Adam*
Loss Function	*MSE*
Warm up Samples	*2000*
Train/Validation Split	*75%/25%*

TABLE V
DATASET MANAGEMENT

Training	*Validation*	*Test*
Batch #0/1 *(12993 samples)*	*Batch #0/2* *(4331 samples)*	*Batch #1 #2 #3* *(~27000 samples)*

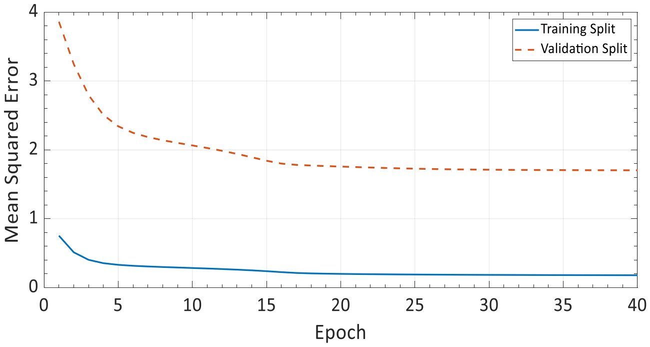

Fig. 6. Network training results.

results concerning the best set of the previous rounds. The optimal parameters obtained at the end of the optimization process are shown in Table IV and have been used for both LSTM and FRNN training.

More in detail, after the operation of normalization described in Section II, the data have been organized, with the aid of an appropriate Python script, in such a way to respect the shape foreseen by the input layer of the neural network.

At the end of the training operations, the best results have brought back a validation loss, mean squared error, equal to 1.70, reached on the 35th epoch.

The summary of the dataset split can be seen in Table V, while the training trend and the loss and validation loss parameters can be seen in Fig. 6.

After the training, the model was compiled and used to predict and estimate soil relative humidity data in the inference phase.

V. RESULTS AND DISCUSSION

A. Algorithm Comparison

Following the neural network training, it was decided to test its results on three sets of measurements taken at different locations. In particular, the three sets covered a period of at least nine months, with data acquired between 2020 and 2021. As a result, the variability of the data is very high, as they covered periods of cold and heat, related to the change of season between winter and summer, and also between drought and large rainfall, as evidenced by the variability of precipitation.

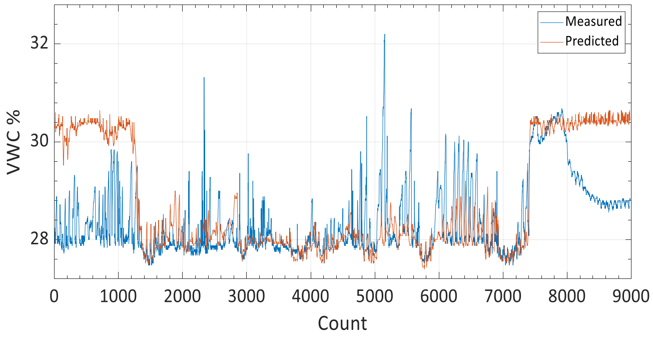

Fig. 7. Extract of the proposed soft sensor's output compared to the actual sensor's response.

These factors predicted relative soil moisture values more challenging. It was, therefore, necessary to set up, as described above, a long warmup period of about 2000 samples, equal to about 34 days. Following the inference operation, it was also necessary to de-normalize the resulting data by reversing (7) using the same mean and standard deviation values.

The results of this evaluation were then compared and visualized to verify consistency with actual soil moisture data to understand the true capability of the system to replace the relative sensor.

As an example, a batch of data having a time span of about 180 days is reported, as an example, in Fig. 7. In particular, the predicted values, in orange, and the measured values, in blue, are compared.

The same operation was performed with the algorithm chosen as a benchmark and described previously. More in detail, the dataset used for training the neural network has been used to derive the regression model, while the three sets of measurements used during the testing phase of the LSTM were also used to test the other models. The measured values versus the values resulting from the model were then plotted, and the results are reported in Fig. 8. For the validation of the model, a cross-validation fold on four partitions of the dataset was also prepared to avoid overfitting the model.

In order to choose which of the three regression trees (fine, medium, or coarse) was the best, the root mean square error (RMSE), calculated as in (12), was evaluated for each test set of measures

$$\mathrm{RMSE} = \sqrt{\frac{\sum_{i=1}^{N}(\hat{y}_i - y_i)^2}{N}} \tag{12}$$

where N is the number of samples analyzed, $\hat{y}_i$ is the estimated value, and y_i is the real value. The results of the RMSE estimation are reported in Table VI.

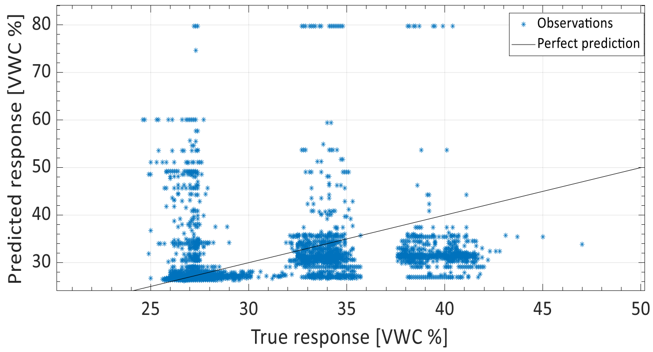

Fig. 8. Medium regression tree algorithm. True values versus predicted sensor's output.

TABLE VI

COMPARISON OF THE RMSE BETWEEN THE PROPOSED LSTM AND THREE DIFFERENT REGRESSION TREES

Model	*Batch #1*	*Batch #2*	*Batch #3*
Fine Tree	*0.64*	*3.44*	*0.91*
Medium Tree	*0.70*	*3.41*	*0.92*
Coarse Tree	*0.59*	*3.39*	*0.84*
NARX	*0.51*	*2.48*	*0.25*
FRNN	*0.22*	*2.51*	*0.26*
LSTM	***0.15***	***0.92***	***0.21***

What stands out from the table is that the coarse tree turns out to be the optimal multivariate regression model due to the smaller mean square error.

Similarly, the RMSE has also been evaluated considering the neural network prediction results since accuracy is not an employable parameter for regression tasks. For comparison, the RMSE of the LSTM approach has been calculated using the same batches of data implemented in the case of the regression tree, NARX, and FRNN. The resulting RMSE values are also reported in the final row of Table VI using bold characters. Fig. 9 shows a batch of true responses versus predicted responses for the LSTM approach. What stands out from the plot is that the results of Table VI are confirmed. Furthermore, the values predicted by the LSTM network resulted in being closer to the bisector (i.e., perfect prediction) than those related to the regression tree method (as in Fig. 8).

Given the models' results and their performances presented in synthetic terms using the RMSE, it can therefore be concluded that the proposed method based on the LSTM network has provided better results on all three batches of data used for testing. In contrast, the NARX network and the FRNN achieved similar performances, better than the multivariate regression trees but worse than the LSTM.

The downside concerns training times: while inference times turn out to be small compared to the dynamics of the phenomenon under investigation (∼1 s), training times can be considerably longer depending on the type of hardware used. In fact, deep neural networks require longer training times than other approaches, which can be cut down by using graphics accelerators and more powerful hardware. In particular, the LSTM has a better RMSE and a significantly higher training time at the same epochs than the FRNN. This is mainly due to the higher architectural complexity of the LSTM unit in comparison with the simple feedback neuron of the FRNN. A laptop without graphics acceleration has been used for this article, and the results are shown in Table VII.

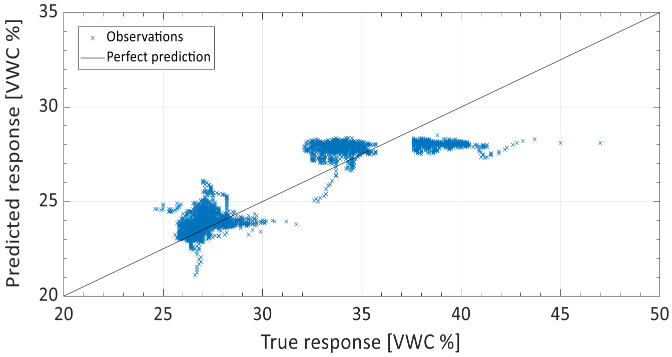

Fig. 9. Proposed LSTM approach. True values versus predicted sensor's output.

TABLE VII

TRAINING TIME COMPARISON

Regression tree	*NARX*	*FRNN*	*LSTM*
~10 sec	*~10 sec*	*~400 sec*	*~2400 sec*

B. Performance Estimation

In order to pursue the main purpose of the work, i.e., to design a soft sensor starting from the technique exposed and tested so far, it was necessary to perform a more in-depth analysis of the predictions in comparison with the ground truth, measured with the appropriate soil moisture sensor, to make calibration of the entire measurement chain.

In a first step, the characteristics of the error committed by the neural network in estimating the relative humidity of the soil were evaluated in more detail. An in-depth analysis of how the average relative error varies across the dataset was then performed for each test batch employed.

For this purpose, it was decided to use a function defined as sliding occurrence error (SOE). This function allows estimating the probability distribution of the average relative error in the form of a survival function, as in the following equation:

$$S(t) = P(\{T > t\}) = \int_t^{\infty} f(u)du = 1 - F(t) \qquad (13)$$

where T is a continuous random variable with cumulative distribution function $F(t)$ on the interval $[0, \infty)$.

Then, the average relative error was calculated using a sliding window on the dataset, according to (14). L_s specifies the width, in terms of the number of samples, of the sliding window, y_p the predicted value, and y_m the measured value

$$E_{\text{mean},L}(i) = \frac{1}{L_S} \sum_{k=0}^{L_S-1} \left| \frac{y_p(i-k) - y_m(i-k)}{y_m(i-k)} \right|. \qquad (14)$$

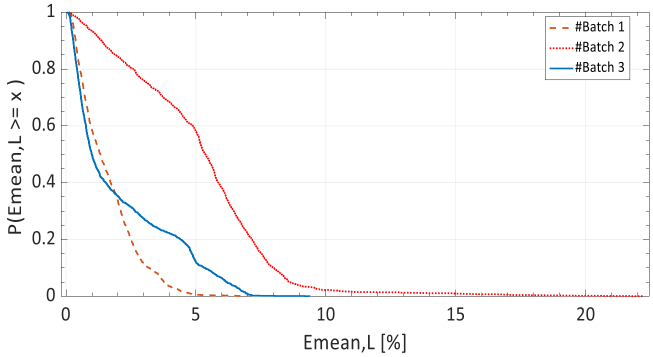

Fig. 10. Comparison of SOE of the three testing data batches.

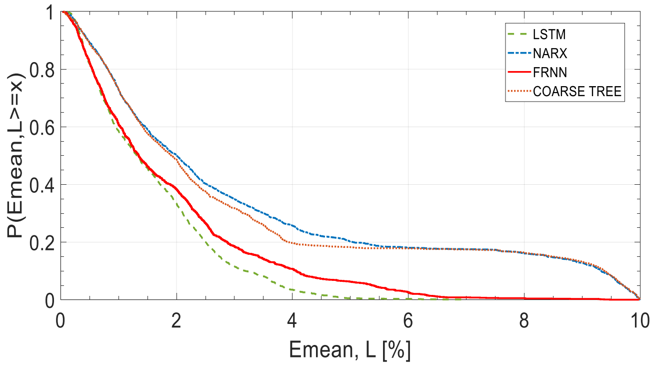

Fig. 11. Comparison of SOE considering the proposed approach and three other state-of-the-art methods.

The SOE curve plots the mean relative deviation $E_{\text{mean},L}$ on the x-axis and the corresponding relative occurrences in the moving window of the regression error on the y-axis.

To improve the prediction accuracy, further optimization has been carried out. First, the length of the sliding window and the overlap between different windows have been parametrized. Following this analysis, a 30-sample wide sliding window was chosen, with an overlap between windows of 80%. The latter corresponds to 24 samples. The optimized results are shown in Fig. 10.

In particular, the first batch of data is the one that obtained the lowest error, as the average relative error of almost all observation windows was confined within 5%. On the other hand, batch three and especially batch two reported larger mean relative errors even though the probability of it being greater than 0.08 is still less than 10%.

The last analysis has been about comparing the SOE obtained for the best batch of data (the first one) predicted using the LSTM network, the coarse regression tree method (i.e., the regression tree method that ensures the best results for the same batch of data), the NARX and the FRNN. The results of this comparison are illustrated in Fig. 11. Also, in this case, we have the substantial confirmation of the results reported previously, as the probability level associated with the average error of the sliding window, calculated on the output data of the LSTM network, turns out to be lower than that of the regression tree on the whole interval under examination.

At the same time, FRNN performed slightly worse than LSTM, and NARX had almost the same performance as the regression tree. These tests, therefore, verified the improvements of using the LSTM network for soft sensing over the use of traditional machine learning algorithms, particularly over the regression tree and NARX.

VI. Conclusion

This work deals with a soft sensing solution to improve the performance of a WSN for smart farming applications. The developed WSN is based on ten autonomous sensor nodes equipped with an air temperature and humidity sensor, a soil temperature sensor, a soil moisture sensor, and a radiation sensor. Due to several limitations in using a physical sensor for measuring the relative humidity of the soil, this work introduces a soft sensing algorithm based on a DL approach to implement a virtual soil moisture sensor. The virtual sensor has proven to be a remarkable advantage despite the classical physical sensor, allowing a cost reduction, an increment in system reliability, and an easier installation and maintenance of the node. Compared with NARX, FRNN, and multivariate regression tree networks, the proposed technique's main downside lies in the long training time. However, this problem can be solved by using more powerful hardware at the time of training since, in the inference phase, the time required is significantly lower and in line with the dynamics of the observed phenomenon. To test and validate the performances of the proposed LSTM-based soft sensing approach, a comparison with three alternatives of the classical multivariate regression tree algorithm has been presented. The results show the goodness of estimation of the proposed virtual sensor regarding RMSE and SOE.

References

[1] V. Nasir and F. Sassani, "A review on deep learning in machining and tool monitoring: Methods, opportunities, and challenges," *Int. J. Adv. Manuf. Technol.*, vol. 115, no. 9, pp. 2683–2709, May 2021.

[2] L.-R. Jácome-Galarza, M.-A. Realpe-Robalino, J. Paillacho-Corredores, and J.-L. Benavides-Maldonado, "Time series in sensor data using state-of-the-art deep learning approaches: A systematic literature review," in *Communication, Smart Technologies and Innovation for Society*. Singapore: Springer, 2022, pp. 503–514.

[3] A. Khanna and S. Kaur, "Evolution of Internet of Things (IoT) and its significant impact in the field of precision agriculture," *Comput. Electron. Agricult.*, vol. 157, no. 1, pp. 218–231, 2019.

[4] M. Catelani, L. Ciani, A. Bartolini, C. Del Rio, G. Guidi, and G. Patrizi, "Reliability analysis of wireless sensor network for smart farming applications," *Sensors*, vol. 21, no. 22, p. 7683, Nov. 2021.

[5] M. Lezoche, J. E. Hernandez, M. D. M. E. Alemany Díaz, H. Panetto, and J. Kacprzyk, "Agri-food 4.0: A survey of the supply chains and technologies for the future agriculture," *Comput. Ind.*, vol. 117, May 2020, Art. no. 103187.

[6] T. Ojha, S. Misra, and N. S. Raghuwanshi, "Wireless sensor networks for agriculture: The state-of-the-art in practice and future challenges," *Comput. Electron. Agric.*, vol. 118, pp. 66–84, Oct. 2015.

[7] R. Rayhana, G. G. Xiao, and Z. Liu, "Printed sensor technologies for monitoring applications in smart farming: A review," *IEEE Trans. Instrum. Meas.*, vol. 70, pp. 1–19, 2021.

[8] K. Alibabaei *et al.*, "A review of the challenges of using deep learning algorithms to support decision-making in agricultural activities," *Remote Sens.*, vol. 14, no. 3, p. 638, Jan. 2022.

[9] K. Alibabaei, P. D. Gaspar, and T. M. Lima, "Crop yield estimation using deep learning based on climate big data and irrigation scheduling," *Energies*, vol. 14, no. 11, p. 3004, May 2021.

[10] K. Alibabaei, P. D. Gaspar, E. Assunção, S. Alirezazadeh, and T. M. Lima, "Irrigation optimization with a deep reinforcement learning model: Case study on a site in Portugal," *Agricult. Water Manage.*, vol. 263, Apr. 2022, Art. no. 107480.

[11] M. Sami *et al.*, "A deep learning-based sensor modeling for smart irrigation system," *Agronomy*, vol. 12, no. 1, p. 212, Jan. 2022.

[12] T. Moon, H. Y. Choi, D. H. Jung, S. H. Chang, and J. E. Son, "Prediction of CO_2 concentration via long short-term memory using environmental factors in greenhouses," *Horticultural Sci. Technol.*, vol. 38, no. 2, pp. 201–209, 2004.

[13] S. M. Hassan, M. Jasinski, Z. Leonowicz, E. Jasinska, and A. K. Maji, "Plant disease identification using shallow convolutional neural network," *Agronomy*, vol. 11, no. 12, p. 2388, Nov. 2021.

[14] A. Picon, A. Alvarez-Gila, M. Seitz, A. Ortiz-Barredo, J. Echazarra, and A. Johannes, "Deep convolutional neural networks for mobile capture device-based crop disease classification in the wild," *Comput. Electron. Agricult.*, vol. 161, pp. 280–290, Jun. 2019.

[15] G. Hu, H. Wu, Y. Zhang, and M. Wan, "A low shot learning method for tea leaf's disease identification," *Comput. Electron. Agricult.*, vol. 163, Aug. 2019, Art. no. 104852.

[16] S. Coulibaly, B. Kamsu-Foguem, D. Kamissoko, and D. Traore, "Deep neural networks with transfer learning in millet crop images," *Comput. Ind.*, vol. 108, pp. 115–120, Jun. 2019.

[17] D. Shadrin, A. Menshchikov, A. Somov, G. Bornemann, J. Hauslage, and M. Fedorov, "Enabling precision agriculture through embedded sensing with artificial intelligence," *IEEE Trans. Instrum. Meas.*, vol. 69, no. 7, pp. 4103–4113, Jul. 2020.

[18] Z. Ünal, "Smart farming becomes even smarter with deep learning—A bibliographical analysis," *IEEE Access*, vol. 8, pp. 105587–105609, 2020.

[19] V. Meshram, K. Patil, V. Meshram, D. Hanchate, and S. D. Ramkteke, "Machine learning in agriculture domain: A state-of-art survey," *Artif. Intell. Life Sci.*, vol. 1, Dec. 2021, Art. no. 100010.

[20] Q. Tan, X. Han, M. Zheng, and J. Tjong, "Neural network soft sensors for gasoline engine exhaust emission estimation," *J. Energy Resour. Technol.*, vol. 144, no. 8, Nov. 2021.

[21] X. Jiang and Z. Ge, "Augmented multidimensional convolutional neural network for industrial soft sensing," *IEEE Trans. Instrum. Meas.*, vol. 70, pp. 1–10, 2021.

[22] M. G. Xibilia, M. Latino, Z. Marinkovic, A. Atanaskovic, and N. Donato, "Soft sensors based on deep neural networks for applications in security and safety," *IEEE Trans. Instrum. Meas.*, vol. 69, no. 10, pp. 7869–7876, Oct. 2020.

[23] D. Capriglione, M. Carratu, A. Pietrosanto, and P. Sommella, "Soft sensors for instrument fault accommodation in semiactive motorcycle suspension systems," *IEEE Trans. Instrum. Meas.*, vol. 69, no. 5, pp. 2367–2376, May 2020.

[24] M. Carratù, C. Liguori, A. Pietrosanto, M. O'Nils, and J. Lundgren, "Data fusion for timber bundle volume measurement," in *Proc. IEEE Int. Instrum. Meas. Technol. Conf. (I2MTC)*, May 2019, pp. 1–6.

[25] P. Suebsombut, A. Sekhari, P. Sureephong, A. Belhi, and A. Bouras, "Field data forecasting using LSTM and bi-LSTM approaches," *Appl. Sci.*, vol. 11, no. 24, p. 11820, Dec. 2021.

[26] N. A. N. M. Adib and S. Daliman, "Conceptual framework of smart fertilization management for oil palm tree based on IoT and deep learning," in *Proc. IOP Conf., Earth Environ. Sci.*, Aug. 2021, vol. 842, no. 1, Art. no. 012072.

[27] K. Pothuganti, B. Sridevi, and P. Seshabattar, "IoT and deep learning based smart greenhouse disease prediction," in *Proc. Int. Conf. Recent Trends Electron., Inf., Commun. Technol. (RTEICT)*, Aug. 2021, pp. 793–799.

[28] R. Liao, S. Zhang, X. Zhang, M. Wang, H. Wu, and L. Zhangzhong, "Development of smart irrigation systems based on real-time soil moisture data in a greenhouse: Proof of concept," *Agricult. Water Manage.*, vol. 245, Feb. 2021, Art. no. 106632.

[29] H. Liu, Y. Yang, X. Wan, J. Cui, F. Zhang, and T. Cai, "Prediction of soil moisture and temperature based on deep learning," in *Proc. IEEE Int. Conf. Artif. Intell. Comput. Appl. (ICAICA)*, Jun. 2021, pp. 46–51.

[30] C. Saad Hajjar, C. Hajjar, M. Esta, and Y. Ghorra Chamoun, "Machine learning methods for soil moisture prediction in vineyards using digital images," in *Proc. E3S Web Conf.*, vol. 167, Apr. 2020, p. 02004.

[31] U. Acharya, A. L. M. Daigh, and P. G. Oduor, "Machine learning for predicting field soil moisture using soil, crop, and nearby weather station data in the red river valley of the north," *Soil Syst.*, vol. 5, no. 4, p. 57, Sep. 2021.

[32] S. Prakash, A. Sharma, and S. S. Sahu, "Soil moisture prediction using machine learning," in *Proc. 2nd Int. Conf. Inventive Commun. Comput. Technol. (ICICCT)*, Apr. 2018, pp. 1–6.

[33] M. ElSaadani, E. Habib, A. M. Abdelhameed, and M. Bayoumi, "Assessment of a spatiotemporal deep learning approach for soil moisture prediction and filling the gaps in between soil moisture observations," *Frontiers Artif. Intell.*, vol. 4, Mar. 2021, Art. no. 636234.

[34] A. Vyas and S. Bandyopadhyay, "Dynamic structure learning through graph neural network for forecasting soil moisture in precision agriculture," 2020, *arXiv:2012.03506*.

[35] C. Kucuk, D. Birant, and P. Yildirim Taser, "An intelligent multi-output regression model for soil moisture prediction," in *Proc. Int. Conf. Intell. Fuzzy Syst.*, 2022, pp. 474–481.

[36] Y. Cai, W. Zheng, X. Zhang, L. Zhangzhong, and X. Xue, "Research on soil moisture prediction model based on deep learning," *PLoS ONE*, vol. 14, no. 4, Apr. 2019, Art. no. e0214508.

[37] L. Ciani, M. Catelani, A. Bartolini, G. Guidi, and G. Patrizi, "Influence of raised ambient temperature on a sensor node using step-stress test," *IEEE Trans. Instrum. Meas.*, vol. 69, no. 12, pp. 9549–9556, Dec. 2020.

[38] L. Ciani, M. Catelani, A. Bartolini, G. Guidi, and G. Patrizi, "Design optimisation of a wireless sensor node using a temperature-based test plan," *ACTA IMEKO*, vol. 10, no. 2, p. 37, Jun. 2021.

[39] R. May, G. Dandy, and H. Maier, "Review of input variable selection methods for artificial neural networks," in *Artificial Neural Networks-Methodological Advances and Biomedical Applications*. London, U.K.: Rijeka, Croatia: IntechOpen, 2011. [Online]. Available: https://www.intechopen.com/chapters/14882.

[40] H. Luo, F. Cheng, H. Yu, and Y. Yi, "SDTR: Soft decision tree regressor for tabular data," *IEEE Access*, vol. 9, pp. 55999–56011, 2021.

[41] S. Jamil, T. Mohd, S. Masrom, and N. Ab Rahim, "Machine learning price prediction on green building prices," in *Proc. IEEE Symp. Ind. Electron. Appl. (ISIEA)*, Jul. 2020, pp. 1–6.

[42] M. Z. Alom *et al.*, "A state-of-the-art survey on deep learning theory and architectures," *Electronics*, vol. 8, no. 3, p. 292, Mar. 2019.

[43] G. Peter and M. Matskevichus, "Hyperparameters tuning for machine learning models for time series forecasting," in *Proc. 6th Int. Conf. Social Netw. Anal., Manage. Secur. (SNAMS)*, Oct. 2019, pp. 328–332.

Gabriele Patrizi (Member, IEEE) received the bachelor's degree (*cum laude*) in electronic and telecommunications engineering, the master's degree (*cum laude*) in electronic engineering, and the Ph.D. degree in industrial and reliability engineering from the University of Florence, Florence, Italy, in 2015, 2018, and 2022, respectively.

He is currently a Post-Doctoral Research Fellow in the field of instrumentation and measurement and an Adjunct Lecturer of electric measurements with the University of Florence. His current research interests include life cycle reliability of complex systems, condition monitoring for fault diagnosis of electronics, safety instrumented systems, data-driven prognostic, and health management.

Alessandro Bartolini (Graduate Student Member, IEEE) received the B.S. degree in electronic and telecommunications engineering and the M.S. degree in electronics engineering from the University of Florence, Florence, Italy, in 2015 and 2018, respectively, where he is currently pursuing the Ph.D. degree in smart industry.

He participated to the NATO-funded project G5014 "Holographic and Impulse Subsurface Radar for Landmine and improvised explosive device (IED) Detection." His current research interests include hardware and software design for precision smart farming.

Lorenzo Ciani (Senior Member, IEEE) received the M.S. degree in electronic engineering and the Ph.D. degree in industrial and reliability engineering from the University of Florence, Florence, Italy, in 2005 and 2009, respectively.

He is currently an Associate Professor with the Department of Information Engineering, University of Florence. He has authored or coauthored more than 190 peer-reviewed journal and conference papers. His current research interests include system reliability, availability, maintainability and safety, reliability evaluation test and analysis for electronic systems and devices, fault detection and diagnosis, and electrical and electronic instrumentation and measurement.

Dr. Ciani is a member of the IEEE IMS TC-32 Fault Tolerant Measurement Systems. He is an Associate Editor-in-Chief of the IEEE TRANSACTIONS ON INSTRUMENTATION AND MEASUREMENT and an Associate Editor of IEEE ACCESS. He received the 2015 IEEE I&M Outstanding Young Engineer Award for "his contribution to the advancement of instrumentation and measurement in the field of reliability analysis".

Vincenzo Gallo (Graduate Student Member, IEEE) was born in 1997. He received the B.S. and the M.S degrees (Hons.) in electronic engineering from the University of Salerno, Fisciano, Italy, in 2018 and 2020, respectively, where he is currently pursuing the Ph.D. degree in industrial engineering.

His current research interests include the development of deep learning (DL) techniques for image-based measurements and the development of novel measurement techniques based on innovative machine learning approaches.

Mr. Gallo is actually a member of the I&M Society and Italian Group of Electric and Electronic Measurement "GMEE," Italy.

Paolo Sommella (Member, IEEE) was born in Salerno, Italy, in 1979. He received the M.S. degree in electronic engineering and the Ph.D. degree in information engineering from the University of Salerno, Fisciano, Italy, in 2004 and 2008, respectively.

In 2015, he joined the Department of Industrial Engineering (DIIn), University of Salerno, as an Assistant Professor of electrical and electronic measurements. His current research interests include instrument fault detection and isolation, measurement in software engineering, and biomedical image processing.

Marco Carratù (Member, IEEE) was born in Brugg, Switzerland, in 1989. He received the M.S. degree in electronic engineering and the Ph.D. degree from the University of Salerno, Fisciano, Italy, in 2015 and 2019, respectively.

He is currently with the Department of Industrial Engineering, University of Salerno, as a Research Fellow of electronic measurements. His current research interests include instrument fault detection and isolation, sensor data fusion, digital signal processing for advanced instrumentation, artificial neural network (ANN), and embedded system.