On the impact of the stem electrical impedance in neural network algorithms for plant monitoring applications

On the impact of the stem electrical impedance in neural network algorithms for plant monitoring applications

Mattia Barezzi*, Federico Cum*, Umberto Garlando*, Maurizio Martina* and Danilo Demarchi*
*Department of Electronics and Telecommunications (DET), Politecnico di Torino, Torino, Italy

Email: umberto.garlando@polito.it

***Abstract*—Smart agriculture offers an environmental-friendly path with respect to unsustainable farming that depletes the nutrients in the soil leading to a persistent degradation of ecosystems caused by population growth. Artificial Intelligence (AI) can help mitigate this issue by predicting plant health status to reduce the use of chemicals and optimize water usage. This paper proposes a custom framework to train neural networks and a comparison among different models to point out the impact and the importance of the stem electrical impedance in addition to environmental parameters for plant monitoring applications. In particular, the paper demonstrates how stem electrical impedance improves the accuracy of the proposed neural network application for plant status classification. The data set is composed of electrical impedance spectra and environmental data acquired on four tobacco plants for a two-month-long experiment. In this paper, we describe the acquisition system architecture, the feature composition of the data set, a general overview of the developed framework, and the training of the neural networks showing the different results considering both the stem impedance and the environmental parameters.**

***Index Terms*—Smart agriculture, neural networks, plant's health status**

I. Introduction & background

Food security is challenged by human population growth and the reduction of arable lands [1]. Therefore, smart agriculture is trying to improve food production by integrating sensors and automation technology with the farmer's knowledge. Sensors typically monitor the environment's physical quantities. For example, air humidity, temperature, and light intensity are commonly monitored environmental parameters. However, the soil is also crucial: soil moisture or volumetric water content are typically sensed. Besides, monitoring the plant itself is also important. In this regard, leaf temperature [2], sap flow [3], and stem impedance spectroscopy [4] are the most interesting quantities.

The most helpful information is the status of the plant in order to understand if the plant is suffering, for example, due to water stress or the presence of parasitic. Machine learning can be applied in this field, where the data of the environment and the plants can be used to train a model that can be employed to classify the plant status of a living being. As a starting point, data previously collected can be used to verify this approach. Furthermore, it is possible to understand which parameters are important to determine the plant's status. This evaluation could be carried out as a supervised problem (also called classification problem) or unsupervised (also called clusterization problem) but also semi-supervised or as reinforcement learning are possible choices. The first is the optimal choice when a given data set can be labeled. In this way, when a machine is so-called trained, that machine can recognize to what class belongs a specific combination of input data. In the agrifood field, the plant is a complex system where it is possible to recognize the most valuable "top-level" data summarized as plant health status to optimize a crop. For example, coarse modeling could be a plant classification based on leaves condition. Alternatively, a more sophisticated analysis could be performed, for example, monitoring volatile organic compounds emitted by the plants and related to their current status using, for example, optical and electrochemical ethylene sensors or electrochemical and chemo-fluorescent jasmonate sensors [5].

A machine learning application for agriculture could be a binary classification problem (healthy or unhealthy plant) or a more complex multi-class classification where it is possible to evaluate several parameters such as watering stress, correct fertilization, good climate conditions, or, finally, predict the changes of a defined physical quantity based on other parameters. These results are extremely useful in many practical applications, for example, yield prediction, disease detection, weed detection, crop quality, and water and soil monitoring [6].

It is helpful to recall that in the supervised algorithms lie several learning models: k-Nearest Neighbor (k-NN) [7], Naive-Bayes [8], decision tree [9], [10] or random forest, linear and logistic regression, Support Vector Machine (SVM) [11] and neural network algorithms are used in many different fields. Each of them has to be set up in such a way as to adapt the underlying algorithm to the practical problem and obtain the best possible performance. In [6], a comparison among supervised, partially supervised, and unsupervised machine learning algorithms is explained.

Finally, it needs to consider also the practical technological scenario [12]: the chosen algorithm should be implemented on-field, for example, in an IoT (Internet of Things) node where a critical aspect is related to power consumption. For this reason, machine learning algorithms are usually trained on a PC (Personal Computer) or an HPC (High-Performance Computing) cluster, which is the most time-consuming and computational-intensive step. Then, the trained model is imported in a low-cost and low-power microcontroller. In this way, it is also possible to deploy on IoT node fancy models. Fig. 1 shows a possible development flow of the machine learning architecture.

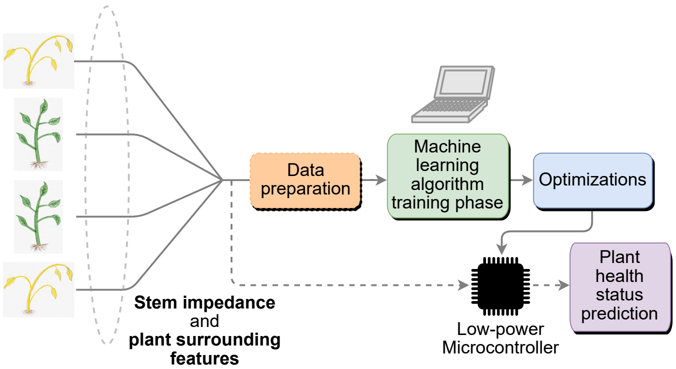


Fig. 1. Block diagram of plants' health status prediction.

The paper is organized as follows: section II describes how the data set for this study was extracted, the general structure of the developed framework, and how the model was implemented. Then, section III shows the obtained results, and section IV draws the conclusions.

II. Proposed Design

A. Data set

A data set is needed to train a model that can be used to predict tobacco plants' health status. A system has been developed to sense plants' stem impedance and their surrounding parameters [13], [14], summarized in Fig. 2.

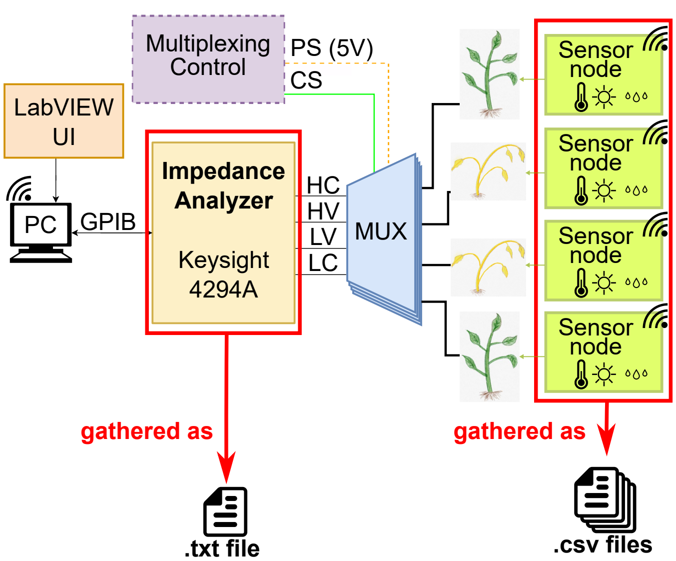


Fig. 2. Block diagram of the acquisition system.

The stem impedance module and phase are measured using an impedance analyzer (Keysight 4294A). The four-probe technique is applied using two Kelvin clips connected to the plant's stem through tiny stainless steel needles. A complete explanation related to impedance spectroscopy in tobacco plants is presented in [15]. The impedance analyzer measures four plants thanks to a multiplexing circuit (MUX) controlled by a Raspberry Pi (Multiplexing Control) that drive Power Supply (PS) and Control Signal (CS) lines. Finally, this appliance was connected to a PC via GPIB (General Purpose Interface Bus), where a LabVIEW program continuously monitors the measurement procedure and stores the impedance spectra. A simple User Interface (UI) has been developed to handle these operations. On the contrary, for the surrounding parameters of each plant, a sensor node was developed as a custom board mounted on the top of a Raspberry Pi Zero W. This sensor node can collect data related to air temperature and humidity, ambient light, and the soil moisture. The impedance is measured at a frequency of 10.145 kHz as the working point that offers the most appreciable variations in tobacco plants [14].

The system samples every one hour all physical quantities and provides a .txt file containing the stem impedance module and phase and four .csv files, one for each analyzed plant. These data were sent to a PC via wireless communication. In this way, four different plants are monitored by the autonomous system, where two of them are watered regularly, and the others are left to water stress. This data set contains information related to the four plants from 24 March 2021 to 4 May 2021 that includes the whole 3516 records.

In addition, a camera is used to monitor the plants: a picture is taken every 15 minutes in such a way as to check the plants' status visually. This task is mainly done to label the data set in the proposed design.

In Fig. 3, a picture taken in April shows plants one and four, plants that suffer water stress, with yellow leaves, a clear indication of a bad plant health status.

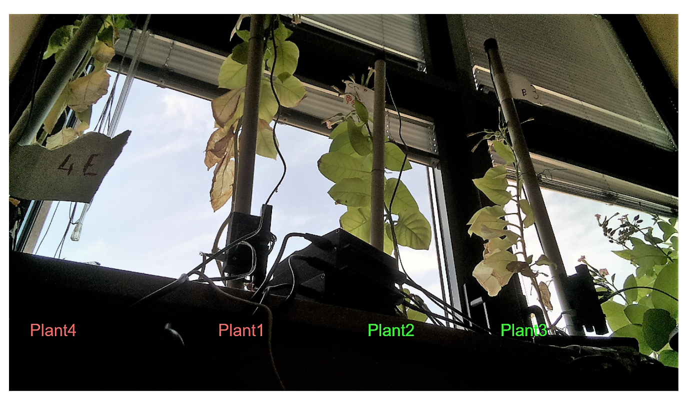


Fig. 3. Picture of the four plants dated 9 April 2021 11:28:34.

The resulting feature set comprises air humidity, air temperature, light intensity, soil moisture, time when the data were sampled, impedance modulus, and impedance phase.

Fig. 4 shows how the model was implemented, considering the described data set.

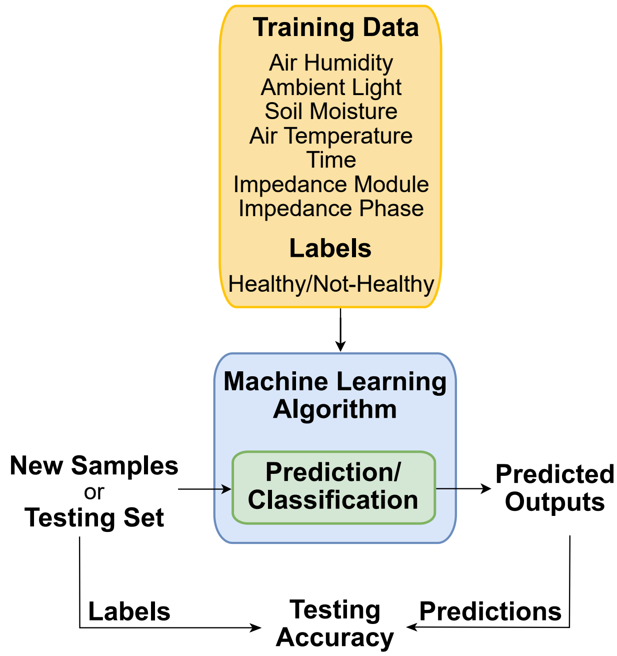


Fig. 4. A machine learning approach using this data set.

A supervised algorithm requires that the data set is labeled: each item of the data set was manually labeled and set up to a logical value whether the plant showed a healthy leaf (green color) or a weakened leaf (yellow color). This is an early qualitative approach to label data to avoid employing expensive sensors. Once the model is trained, it can be used on new samples or in the testing set to evaluate the accuracy figure of merit of the model (testing accuracy).

The following step is data preprocessing: aggregation, sampling, dimensionality reduction, feature subset selection, feature creation, discretization, filtering, or attribute transformation are possible operations to be performed.

For example, in this data set, soil moisture data were clamped to -200 kPa when raw data in the data set showed a lower value than that. This is done because the sensor's datasheet states that the maximum appreciable value is -200 kPa. The column of the time in the input files is in the format $YYYY-MM-DD \quad hh:mm:ss$ where $YYYY$ is the year, MM is the month, DD is the day, hh is the hour, mm is the minute, ss is the second. The data were converted to a numeric format to use this information in the model by exploiting the DateTime library. In particular, only the hours and the minutes are considered. The final value corresponds to the following format number: $hh \times 60 + mm$. Another fundamental step is normalization: this is performed to weigh in the same way all features due to the input quantities having very different ranges. Each feature will converge with a different step size during the training iterations. Therefore, all features have to be in the same range of values: data are normalized to a range from 0 to 1, so it can be expressed with Eq. 1.

$$x' = \frac{x - x_{min}}{x_{max} - x_{min}} \tag{1}$$

Eq. 1 explains x' that is the normalized value, x that is the actual value in the data set, x_{min} and x_{max} that are, respectively, the minimum and maximum values of that feature. Section III shows an example of feature subset selection analysis. This could be an approach to evaluate the importance of acquired features in the training of the model.

Finally, data structures of four plants are concatenated, normalized, shuffled, and separated: 80% of the data is the training set, which will be used to train the model. The remaining 20% of the data is the testing set, which will be used to evaluate the test accuracy. Moreover, starting from 3516 samples, 2813 records compose the training set and 703 records the testing set, respectively.

At this point, a training phase using a machine learning algorithm is performed, and an optimizer is typically used to obtain a model file.

B. Framework

The lack of scientific literature resources about neural networks applied to stem impedance data has required to develop a novel software in such a way as to perform the training process systematically. Moreover, a basic structure was designed and implemented in Python language employing the PyTorch library that provides a wide set of ready-to-use functions to train and test neural networks. This toolchain was chosen since this library is natively object-oriented and it has comprehensive academic support for machine learning applications.

A hierarchical structure was designed to give the possibility to train a model based on a specific hyperparameter configuration or to find the best model based on a sweep over different neural network configurations. Fig. 5 shows a simplified scheme of the developed framework. It was designed starting from the available data set. The main elements of the framework are: settings, feature pre-processing, and dispatcher. The remaining elements in the same figure are the input and output blocks. The former consists of the data set described in II-A. The latter includes the trained output model (in .pth extension) and a set of text files containing helpful information on the model itself.

The settings element is composed of a set of files that are a convenient way to set all the hyperparameters needed to perform neural network training. Besides classical hyperparameters, such as learning rate, number of epochs, batch size, loss function, and optimizer, it is also possible to specify which features have to be employed, the neural network structure or structures to be trained, and evaluation metrics used to evaluate a model.

Feature pre-processing element includes a subset of functions implemented in the PyTorch and DateTime libraries able to manage the data set, making it suitable for the training phase.

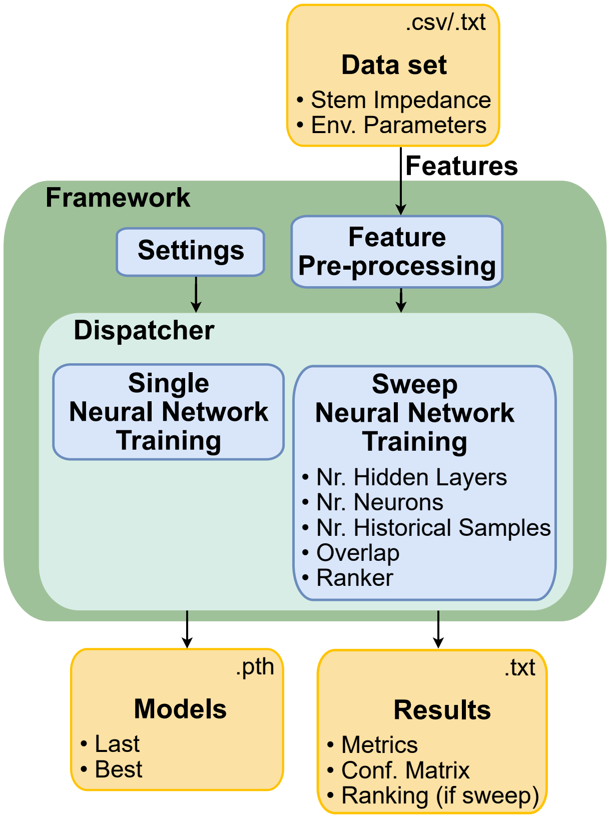


Fig. 5. A general overview of the framework, including input (data set) and output (models and results) elements.

Attribute transformation, normalization, concatenation, and shuffling are only some of the possible choices used in the simulations described in III. The dispatcher is the core of the developed framework and is in charge of using the settings specified by the user to activate the training model process. In particular, two main functions can be chosen: single neural network training and sweep neural network training. In the former case, a single neural network is trained following the settings files. In the latter case, multiple networks are trained, varying some of the hyperparameters. The framework is capable of sweeping over the number of hidden layers and neurons for each hidden layer, the number of past samples used to perform a prediction, and overlap size (subsequent time windows for a prediction). Finally, a structure was implemented to select different ranks for future evaluation with other figures of merit. Each time a model is trained, the last model and the best model according to the accuracy are saved inside a .pth model file. Moreover, accuracy results are saved into text files, and the confusion matrix (a specific table layout that allows visualization of the performance of an algorithm) for each model is evaluated. For the case of the sweep search, the trained models are also ranked based on the test accuracy.

III. Results and discussion

An early set of simulations is needed to find the correct parameters for this data set in the framework. In particular, the correct learning rate, a sufficient long epoch number, the appropriate activation functions, and a suitable optimizer. A good trade-off was found in setting up a learning rate equal to 0.0001, epoch number equal to 300, the 'relu' activation function in the inner layers, 'softmax' activation function in the output layer, and 'Adam' as optimizer. In addition to them, the number of inner layers and their number of neurons should be found. A too-complex network leads to overfitting, and, on the contrary, a too-minimal network leads to underfitting, so the model is too simple to predict in a good way the plant health status. Here, it was found as a reasonable trade-off for a network where there are two hidden layers composed of six neurons, each of them. The number of input neurons depends on the number of considered features employed to train the model. The number of output neurons is fixed to two.

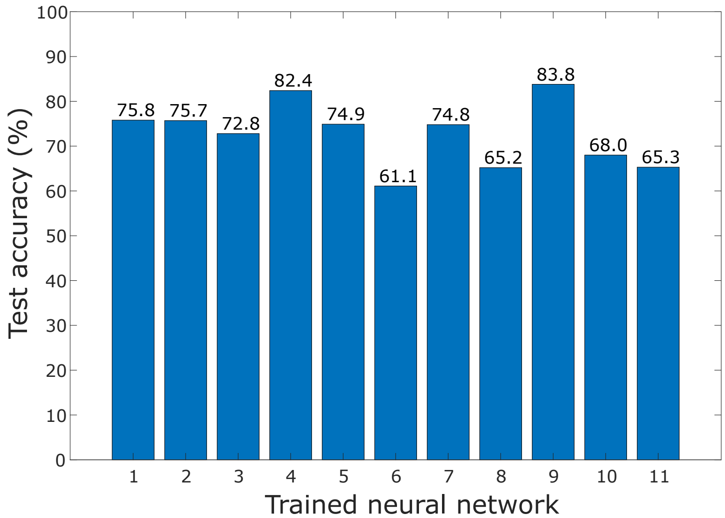
The experiment should highlight the importance of stem impedance as an additional parameter for the proposed neural network training. Moreover, comparing a network composed of seven features (all surrounding features plus stem impedance module and phase) against a network composed of five features (only surrounding features are considered) could not be fair.

For this reason, a comparison among different networks has been performed using the single neural network training feature of the dispatcher, evaluating the best model for each attempt computed by the dispatcher itself. A pair of surrounding features are removed in each network, always considering stem impedance features. Differently, for the last attempt (trial number 11), stem impedance values are not used. In this way, it is possible to prove the importance of stem impedance and the fact that a neural network works better considering both surrounding data and stem impedance. Fig. 6 shows the results in terms of test accuracy of each trained neural network described in the legend on the right side of the figure itself. It is possible to note that neural networks number 3, 6, 8, and 10 show the worst performance (less than 73%) where, in common, there is no soil moisture feature in the training phase. Furthermore, this clearly shows the importance of soil moisture in the model. Another bad performance result appears when the removed features are the impedance modulus and phase (neural network number 11). Moreover, the best networks are related to the presence of both soil moisture and impedance values removing negligible features such as temperature, ambient light, or time. In this way, it has been obtained test accuracy of 82.4% and 83.8%, respectively, in the neural network numbers 4 and 9.

IV. Conclusions

This paper compared trained neural networks using the developed framework that allows great customization in the simulations. This work shows the importance of stem electrical impedance related to plant stress events such as watering events. In addition, it has demonstrated the relevance of soil moisture in predicting tobacco plants' health status.

Future work will be related to implementing alternative solutions to increase the accuracy and evaluate other figures of



Legend

1	No temperature and air humidity
2	No temperature and ambient light
3	No temperature and soil moisture
4	No temperature and time
5	No air humidity and ambient light
6	No air humidity and soil moisture
7	No air humidity and time
8	No ambient light and soil moisture
9	No ambient light and time
10	No soil moisture and time
11	No impedance module and phase

Fig. 6. Trained neural network bar graph comparison.

merit, as for example, the F1 score or Matthews correlation coefficient (MCC) related to the quality of the prediction allowing for the usage of the model on unknown tobacco plants. In addition, it is necessary to collect more data from a broader period and from more plants to train the model, preventing overfitting. Another consideration can be made related to the labeling method used for the plants: actual labeling is based on the color of leaves, but other boundaries could be set up to decide plant health status.

The final goal is to train a network that is possible to be implemented on low-cost and low-power microcontrollers to realize autonomous smart monitoring systems for tobacco plants. Therefore, the predictions may be used to set the optimal conditions for the plants automatically to increase crops and avoid water and fertilizers waste.

Acknowledgment

The authors would like to thank Alessandro Lovesio and Liu Huicai for their work on the definition of the general structure of the custom framework and the performed simulations.

References

[1] A. L. Burrell, J. P. Evans, and M. G. De Kauwe, "Anthropogenic climate change has driven over 5 million km2 of drylands towards desertification." *Nature Communications*, vol. 11, 2020. [Online]. Available: https://doi.org/10.1038/s41467-020-17710-7

[2] L. Yu, W. Wang, X. Zhang, and W. Zheng, "A review on leaf temperature sensor: Measurement methods and application," *9th International Conference on Computer and Computing Technologies in Agriculture (CCTA)*, p. 216–230, Sep 2015.

[3] T. W. Davis, C.-M. Kuo, X. Liang, and P.-S. Yu, "Sap flow sensors: Construction, quality control and comparison," *Sensors*, vol. 12, no. 1, p. 954–971, Jan 2012.

[4] U. Garlando, S. Calvo, M. Barezzi, A. Sanginario, P. Motto Ros, and D. Demarchi, "Ask the plants directly: Understanding plant needs using electrical impedance measurements," *Computers and Electronics in Agriculture*, vol. 193, p. 106707, 2022. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0168169922000242

[5] U. Garlando, L. Bar-On, A. Avni, Y. Shacham-Diamand, and D. Demarchi, "Plants and environmental sensors for smart agriculture, an overview," in *2020 IEEE SENSORS*, 2020, pp. 1–4.

[6] K. G. Liakos, P. Busato, D. Moshou, S. Pearson, and D. Bochtis, "Machine learning in agriculture: A review," *Sensors*, vol. 18, no. 8, 2018. [Online]. Available: https://www.mdpi.com/1424-8220/18/8/2674

[7] A. Nemes, R. Roberts, W. Rawls, Y. Pachepsky, and M. van Genuchten, "Software to estimate -33 and -1500kpa soil water retention using the non-parametric k-nearest neighbor technique," *Environmental Modelling and Software*, vol. 23, no. 2, p. 254–255, 2008.

[8] R. Priya, D. Ramesh, and E. Khosla, "Crop prediction on the region belts of india: A naïve bayes mapreduce precision agricultural model," *2018 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*, 2018.

[9] N. Rochmawati, H. B. Hidayati, Y. Yamasari, W. Yustanti, L. Rakhmawati, H. P. Tjahyaningtijas, and Y. Anistyasari, "Covid symptom severity using decision tree," *2020 Third International Conference on Vocational Education and Electrical Engineering (ICVEE)*, 2020.

[10] S. Salzberg, R. Chandar, H. Ford, S. K. Murthy, and R. White, "Decision trees for automated identification of cosmic-ray hits in hubble space telescope images," *Publications of the Astronomical Society of the Pacific*, vol. 107, p. 279, 1995.

[11] M. Islam, A. Dinh, K. Wahid, and P. Bhowmik, "Detection of potato diseases using image segmentation and multiclass support vector machine," *2017 IEEE 30th Canadian Conference on Electrical and Computer Engineering (CCECE)*, 2017.

[12] M. R. M. Kassim, "Iot applications in smart agriculture: Issues and challenges," in *2020 IEEE Conference on Open Systems (ICOS)*, 2020, pp. 19–24.

[13] U. Garlando, L. Bar-On, P. M. Ros, A. Sanginario, S. Peradotto, Y. Shacham-Diamand, A. Avni, M. Martina, and D. Demarchi, "Towards optimal green plant irrigation: Watering and body electrical impedance," in *2020 IEEE International Symposium on Circuits and Systems (ISCAS)*, 2020, pp. 1–5. [Online]. Available: https://ieeexplore.ieee.org/document/9181290

[14] U. Garlando, L. Bar-On, P. M. Ros, A. Sanginario, S. Calvo, M. Martina, A. Avni, Y. Shacham-Diamand, and D. Demarchi, "Analysis of in vivo plant stem impedance variations in relation with external conditions daily cycle," in *2021 IEEE International Symposium on Circuits and Systems (ISCAS)*, 2021, pp. 1–5. [Online]. Available: https://ieeexplore.ieee.org/document/9401242

[15] L. Bar-On, U. Garlando, M. Sophocleous, A. Jog, P. Motto Ros, N. Sade, A. Avni, Y. Shacham-Diamand, and D. Demarchi, "Electrical modelling of in-vivo impedance spectroscopy of nicotiana tabacum plants," *Frontiers in Electronics*, vol. 2, 2021. [Online]. Available: https://www.frontiersin.org/article/10.3389/felec.2021.753145