

# AI at the Edge: a Smart Gateway for Greenhouse Air Temperature Forecasting

Gaia Codeluppi, Antonio Cilfone, Luca Davoli and Gianluigi Ferrari

*Internet of Things (IoT) Lab*

*Department of Engineering and Architecture*

*University of Parma*

Parma, Italy

{gaia.codeluppi, antonio.cilfone, luca.davoli, gianluigi.ferrari}@unipr.it

**Abstract**—Controlling and forecasting environmental variables (e.g., air temperature) is usually a key and complex part in a greenhouse management architecture. Indeed, a greenhouse inner micro-climate, which is the result of an extensive set of inter-related environmental variables influenced by external weather conditions, has to be tightly monitored, regulated, and, sometimes, forecast. Nowadays, Wireless Sensor Networks (WSNs) and Machine Learning (ML) are two of the most successful technologies to deal with this challenge. In this paper, we discuss how a Smart Gateway (GW), acting as a collector for sensor data coming from a WSN installed in a greenhouse, could be enriched with a Neural Network (NN)-based prediction model allowing to forecast a greenhouse’s inner air temperature. In the case of missing sensor data coming from the WSN, the proposed prediction algorithm, fed with meteorological open data (gathered from the DarkSky repository), is run on the GW in order to predict the missing values. Despite the model is especially designed to be lightweight and executable by a device with constrained capabilities, it can be adopted either at Cloud or at GW level to forecast future air temperature’s values, in order to support the management of a greenhouse. Experimental results show that the NN-based prediction algorithm can forecast greenhouse air temperature with a Root Mean Square Error (RMSE) of 1.50 °C, a Mean Absolute Percentage Error (MAPE) of 4.91%, and a  $R^2$  score of 0.965.

**Index Terms**—Internet of Things, Smart Farming, Edge AI, DNN, Greenhouse Management, Wireless Sensor Network, WSN

## I. INTRODUCTION

From their invention to nowadays, greenhouses, whose aim is to reproduce more favourable conditions for the growing of the inside products [1], have been adopted in order to enable and improve agricultural production anytime and everywhere. The maintenance of a suitable growing habitat is a key (and complex) element in the greenhouse management, since it usually involves to monitor and control a huge number of environmental variables influencing the greenhouse’s internal climate, including, as an example, air and soil temperature and humidity. Moreover, these parameters are usually interrelated and influenced by the meteorological conditions external to the greenhouse (and denoted as “external variables”), such as wind speed, solar radiation, temperature, and humidity [2].

With the rise of the Smart Agriculture concept and a consequent transfer of digital technologies to the agricultural sector, the control of greenhouses’ inner variables has become totally

automated. Indeed, the deployment of Internet of Things (IoT)-oriented systems, usually relying on Wireless Sensor Networks (WSNs), simplifies the real-time measurement of (environmental) sensed data, allowing to *locally* process the information to keep the environmental status under controlled conditions. Moreover, data sensed by Sensor Nodes (SNs) should often be forwarded to another (generally more capable) entity, denoted as Gateway (GW). The GW, in turn, can exploit sensor data to perform actuation and, being typically connected to the Internet, forward them to external services, located either at the Edge or in the Cloud [3], [4]. At Cloud level, collected data can be processed and fused with external information retrieved from other sources (e.g., meteorological and historical data) for different purposes such as, for example, to forecast greenhouse’s future variables trend in order to avoid possible dangerous environmental conditions. Nowadays, this last task has been successfully accomplished with Machine Learning (ML) techniques, such as Artificial Neural Networks (ANNs) [5]–[7].

Since the execution of ML algorithms on IoT devices—near the source of (sensor) data—provides notable advantages, such as lowering the network load (thanks to a reduced amount of data forwarded to the Cloud to be processed) and the latency, a hot IoT trend is to move the intelligence (i.e., execution of Artificial Intelligence, AI, algorithms) from the Cloud to the Edge [8]. Since IoT devices often have significantly lower memory, computational, and energy resources than Cloud platforms, *at-the-Edge* algorithms have thus to be carefully designed (e.g., ANN models with reduced number of parameters) [9].

In the context of greenhouses, algorithms executed at the Edge and targeting internal variables’ forecasting, can be adopted with a two-fold purpose. *First*, the greenhouse’s micro-climate can be properly controlled even if data—related to greenhouse’s inner variables to monitor—are not correctly gathered by SNs (e.g., missing sensors’ data) because of, for example, exhausted batteries, since they can be estimated with Edge forecasting algorithms. *Second*, the prediction of environmental parameters’ values allows to schedule *on-time* management tasks, thus preventing inner variables from reaching critical values (e.g., preemptively activating a cooling system to avoid dangerous temperatures).

The purpose of this paper is to present a novel approach aiming to improve greenhouses management and their internal variables control through the adoption of ANNs, Edge Computing and IoT technologies. In detail, we propose to enhance an IoT node, acting as a GW for a WSN installed in a greenhouse and in charge of monitoring its internal air temperature, with “Edge Intelligence.” We denote this GW as “Smart Gateway” (Smart GW). This is expedient to locally forecast the air temperature of the greenhouse, in the absence of sensed data and, on the basis of the obtained results, to regulate the air temperature. In detail, we discuss (i) the development of a constrained device-friendly prediction model, based on a fully-connected ANN, to forecast the air temperature inside a greenhouse, knowing the outside weather conditions; and (ii) the deployment of the ANN model on a real Smart GW, gathering data from SNs (equipped with sensors measuring the air temperature), forwarding them to the Cloud, and adopting the proposed prediction model to locally forecast potential missing sensor data.

The rest of the paper is organized as follows. In Section II, related works addressing the use of ML to forecast the values of a greenhouses internal environmental variables are presented. Section III discusses the methodology adopted to build the ANN-based prediction model, while details on the considered Smart GW and illustrative experimental results are provided in Section IV. Finally, in Section V we draw some conclusions.

## II. RELATED WORKS

From a technological point of view, several efforts have been undertaken in designing approaches and defining systems aiming at simplifying, enhancing and making the regulation and supervision of greenhouse inner variables automatic. More precisely, as discussed in Section I, automatic management of the internal climate of a greenhouse usually includes: (i) a WSN for collecting and monitoring internal variables, based on IoT technologies [10]–[12]; (ii) several control systems, which automatically perform operations to maintain the internal conditions within desirable ranges [13], [14]; and (iii) internal variables’ prediction techniques, in order to forecast their future trends, based on ML-based algorithms, such as ANNs [15], [16].

With regard to ML, the models proposed in literature, based on ANN and targeting the greenhouse domain, differ in terms of internal processing, input and output variables, and type of Neural Network (NN)-based architecture adopted to solve the prediction problem. These models can be classified as (i) time series-oriented and (ii) “pure” ML.

Concerning the *first* class of models, they take advantage of features typical of time series (i.e., data which are sampled periodically and have a time reference, as sensor data), which include trends, seasonality, and correlation between samples which are closest in time. The best results in this field have been obtained by architectures able to discover deep relations between temporally-close data, such as Recurrent Neural Networks (RNNs) [15] and Long-Short Term Memory

(LSTM) Networks [16]. Unfortunately, a drawback of RNNs and LSTMs is the required significant amount of memory and computational power, with respect to other models, such as Convolutional Neural Networks (CNNs) [17]. In detail, in order to predict the value of a variable at a certain time instant  $t$ , these algorithms usually need a fixed number  $n - 1$  of previous observations to be accessible (collected at time instants  $t - 1, t - 2, \dots, t - n$ ), which are not always available, for example, in IoT systems in which many temporally consecutive sensor data can be lost.

On the other side, internal variables’ forecast can be achieved without considering the relation between temporally-close data, but using other input information, such as external weather parameters and/or other relevant variables (correlated with each one) of a greenhouse climate. In this field, notable results for air temperature prediction have been achieved with Radial Basis Function (RBF) networks [6], [7] or with ANN [18].

As a remark, we underline that the aforementioned works propose algorithms meant to be performed in the Cloud, hence by systems with computing and memory capabilities sufficiently high to run the model and store its parameters. Moreover, although the mentioned models show remarkable results in forecasting air temperatures (e.g., with a Root Mean Squared Error, RMSE, value lower than  $1^\circ\text{C}$ ), when developing ML models on IoT nodes, performance optimization has to take into account the computational and memory resources required by the algorithm. Indeed, when the intelligence is moved to an IoT network, where IoT nodes have limited capabilities with respect to the Cloud, usually a balanced trade-off between algorithm’s performance and computational requirements has to be met. This means that, even if their efficiency is typically lower, lightweight prediction models are normally preferable to better-performing heavier ML models (e.g., small ANNs against models with a huge number of hidden layers and parameters).

Since our proposed forecasting algorithm is intended to be deployed on a “constrained device” (the Smart GW), its prediction performance will be compared with those of algorithms proposed in the literature. Obviously, we expect that lightweight features will likely lower the Smart GW’s performance with respect to those of algorithms deployed in the Cloud.

## III. METHODOLOGY

In order to build a fully-connected ANN-based air temperature prediction model, the following steps have been undertaken: (i) collection of relevant data; (ii) data cleaning and pre-processing; (iii) features’ selection; (iv) dataset definition and splitting between training set and test set; (v) definition, training and optimisation of the prediction model; and (vi) execution of the model on a training set.

### A. Data Gathering, Cleaning and Pre-processing

The agricultural data exploited in this paper, collected from two different data sources, are the following: (i) meteorological

data, coming from the DarkSky weather data repository [19], and (ii) air temperature data gathered with the LoRaFarM platform, a Farm-as-a-Service (FaaS) architecture presented in [20], during a 10-month period (from August 2019 to June 2020), in an Italian greenhouse (namely, Podere Campaz [21]).

### B. Feature Engineering

A preliminary analysis, based on the graphical visualisation of sensor’s data collected inside the greenhouse, revealed that the time series associated with the greenhouse’s internal air temperature have a daily seasonality. In other words, the air temperatures of different days show similar trends. Moreover, hourly values of temperatures are influenced by the year’s month and season in which they are measured. This last trend is justified by the fact that during months of the same season, the farmer adopts a common greenhouse rooftop’s opening/closure pattern, in order to prevent high values of air temperature and humidity.

Hence, the above described correlation among air temperature values and *hour*, *month*, and *season* in which data samples are collected (namely, features which can be derived from a time reference) can be exploited in order to extract 9 new features (denoted as *time-based* in the following). Furthermore, beside features related to meteorological data, time-based features are evaluated as potential inputs for the ANN-based prediction model in the feature selection stage.

In detail, new time-based features are created as follows. *First*, a time-based categorical variable (namely, *hour*, *month*, and *season*) is denoted as  $v$  and will assume an integer value  $x \in [0, \dots, T - 1]$ , where  $T$  is the periodicity of the variable  $v$  and is equal to 24, 12, and 4, for hour, month, and season, respectively. For example, for  $v = \textit{season}$ ,  $x \in [0, 1, 2, 3]$  represents a season in [*winter*, *spring*, *summer*, *autumn*].

*Second*, three trigonometric functions of the time-based variables introduced above, based on sine-, cosine-, and 2-argument arctangent transformations, are defined as follows:

$$v_{\sin} = \sin\left(2\pi\frac{x}{T}\right) \quad (1)$$

$$v_{\cos} = \cos\left(2\pi\frac{x}{T}\right) \quad (2)$$

$$v_{\text{atan2}} = \text{atan2}(v_{\sin}, v_{\cos}). \quad (3)$$

In detail, the trigonometric functions of  $v$  in Eqs. (1)-(3) allow to properly encode the aforementioned categorical variables  $v$  (i.e., *hour*, *month*, and *season*) in numerical variables preserving their periodicity. Moreover, this encoding makes the transformed values of  $v$  numerically close to each other when associated with hours, months or seasons close to each other in time. For instance, when  $v = \textit{season}$ ,  $x = 0$  corresponds to winter and  $x = 3$  corresponds to autumn: the two seasons are close to each other but the numerical values identifying them (0 and 3) are not. However, when considering trigonometric transformations according to Eqs. (1)-(3), the transformed values corresponding to 0 and 3 (namely, 0 and  $-1$ ), are closed to each other.

*Finally*, the trigonometric transformations in Eqs. (1)-(3) lead to 9 new features related to hour (denoted

TABLE I  
SELECTED FEATURES AND THEIR CORRELATION WITH AIR TEMPERATURE INTERNAL TO THE GREENHOUSE.

| Meteorological Feature | Correlation $r$ | Time-based Feature      | Correlation $r$ |
|------------------------|-----------------|-------------------------|-----------------|
| Apparent temperature   | 0.896           | $hour_{\cos}$           | -0.470          |
| Dew point              | 0.690           | $month_{\sin}$          | -0.625          |
| Air humidity           | -0.424          | $month_{\cos}$          | -0.266          |
| Air temperature        | 0.900           | $season_{\cos}$         | -0.652          |
| UV index               | 0.659           | $season_{\text{atan2}}$ | 0.263           |

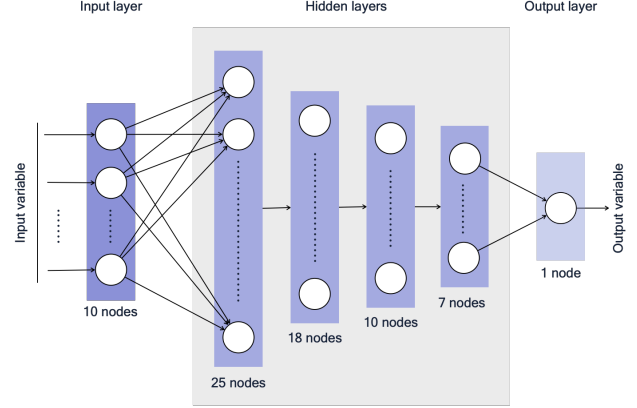


Fig. 1. Layered representation of the proposed fully-connected ANN.

as  $hour_{\sin}$ ,  $hour_{\cos}$ , and  $hour_{\text{atan2}}$ ), month (denoted as  $month_{\sin}$ ,  $month_{\cos}$ , and  $month_{\text{atan2}}$ ), and season (denoted as  $season_{\sin}$ ,  $season_{\cos}$ , and  $season_{\text{atan2}}$ ).

### C. Feature Selection

With the aim of discovering which features are mostly correlated with the greenhouse’s air temperature, a correlation analysis [22] between (i) air temperature samples, collected inside the greenhouse, (ii) meteorological data, and (iii) time-based features (added in the feature engineering stage) has been performed. Then, the resulting features have been selected as inputs variables for the ANN-based prediction model. More precisely, the features, whose absolute correlation value (denoted as  $r$ ) with the air temperature is lower than 0.25, have been discarded. Our experimental results, obtained by performance evaluation of the forecasting model—in terms of RMSE with different numbers of features (namely, from 26 to 1, discarded in descending order for  $r$ )—show that the threshold value of 0.25 is a fair trade-off between the amount of used input features (no more than 10) and prediction performance ( $\text{RMSE} \leq 1.50^\circ\text{C}$ ). Indeed, the larger the number of discarded variables with low correlation with the internal air temperature (in the algorithm case-study), the poorer the prediction performance. The 10 selected features and the corresponding values of the correlation  $r$  with air temperature, are summarized in Table I. As can be seen, there exists a strong positive correlation between the air temperature inside and outside the greenhouse. This is justified by the fact that climate inside the greenhouse is only marginally regulated by internal actuators. Indeed, the air temperature (and humidity) values inside the greenhouse are uniquely controlled by opening or closing the greenhouse’s rooftop.

Moreover, during warm months (May-August), the greenhouse remains open during all daylight hours. Due to these reasons, the internal micro-climate is highly influenced by external weather conditions and, thus, is strongly correlated with the air temperature outside the greenhouse.

#### D. Deployed NN-based Model

After the feature selection stage, a 5346-sample dataset is obtained, in which each sample is composed of: (i) a 10-dimensional vector of input variables, corresponding to the features shown in Table I; and (ii) an output variable, corresponding to the air temperature sensor's value collected inside the greenhouse. Then, the dataset is randomly split into a training set and a test set, with a proportion of 3:1, while the input variables are standardized (i.e., re-scaled in order to obtain variables with a mean value equal to 0 and a standard deviation equal to 1).

In detail, the prediction problem is modeled with a *pure* ML (and not a time series-oriented) approach, in order to better deal with consecutive missing sensor data's prediction. A lightweight—in terms of required storage space and computing requirements—learning architecture, based on a fully-connected ANN with 4 hidden layers and 1018 parameters, has been adopted. As outlined in Section II, another attractive architecture to be considered for air temperature prediction is based on RBF networks. Since this kind of networks usually adopt a weighted sum of non-linear functions (e.g., Gaussian functions) as neurons' activation functions, their execution is generally computationally more expensive than the weighted sum performed by a typical ANN node. On the basis of this observation and taking into account the deployment on a constrained IoT device, the approach based on RBF networks has been discarded.

The architecture of the designed ANN, which has been trained with a Back-Propagation (BP) algorithm and RMSE function as loss function, is shown in Fig. 1. Finally, a  $k$ -fold cross-validation technique (with  $k = 4$ ) has been adopted in order to find the best dataset split, i.e., the partition of samples between training set and test set, which allows the proposed ANN algorithm to reach the best estimation quality (in terms of RMSE).

#### E. Performance Evaluation Criteria

The performance of the proposed forecasting model is evaluated considering three common metrics, namely (i) RMSE, (ii) Mean Absolute Percentage Error (MAPE), and (iii) coefficient of determination ( $R^2$ ), which can be expressed as follows,

respectively [18]:

$$RMSE = \sqrt{\frac{\sum_{j=1}^n (d_j - p_j)^2}{n}} \quad (4)$$

$$MAPE = \frac{1}{n} \sum_{j=1}^n \left| \frac{d_j - p_j}{d_j} \right| \cdot 100 \quad (5)$$

$$R^2 = \left[ \frac{\sum_{j=1}^n (d_j - \bar{d})(p_j - \bar{p})}{\sqrt{\sum_{j=1}^n (d_j - \bar{d}) \sum_{j=1}^n (p_j - \bar{p})}} \right]^2 \quad (6)$$

where:  $d_j$  and  $p_j$  are the actual and forecast values of the  $j$ -th sample;  $\bar{d}$  and  $\bar{p}$  are the average values of the real and forecast value of the  $j$ -th sample element; and  $n$  is the number of samples predicted. The lower RMSE and MAPE, and the higher  $R^2$  (i.e., close to 1), the better the algorithm performance.

## IV. EXPERIMENTAL RESULTS

The prediction performance of the proposed ANN-based model has been evaluated on a test set of 1336 samples (collected from August 2019 to June 2020) using the three metrics indicated in Subsection III-E. Experimental results highlight that the AI model can predict air temperatures with a RMSE, MAPE, and  $R^2$  equal to 1.50 °C, 4.91%, and 0.965, respectively. In other words, this means that the air temperature can be forecast with a standard deviation of 1.50 °C with respect to the actual value. If compared with the models proposed in the literature (recalled in Section II), which allow to predict air temperature values with a RMSE < 1 °C, then the RMSE of our proposed model is slightly higher (in the range of 0.5-1.0 °C [7], [16]) or similar [15]. Considering the  $R^2$  metric, the obtained value (i.e., 0.965) is similar to the results outlined in [16], but higher than the score in [15]. Nevertheless, the obtained prediction performance is satisfactory, taking into account the accuracy of the environmental sensor adopted to measure air temperature (i.e.,  $\pm 1$  °C) and the type of application (namely, greenhouse's internal air temperature monitoring). Moreover, the slight performance degradation is justified by the fact that the forecasting model is more lightweight and less resource-intensive than those proposed in the literature.

In detail, the forecast air temperatures and data collected from sensors during a 5-day period in August 2019, are shown in Fig. 2. Moreover, the performance of the proposed prediction model on the test set, in terms of difference between the collected air temperature data and the predicted ones, is shown in Fig. 3. As can be seen from Fig. 2 and Fig. 3, there is an acceptable agreement between actual sensor data and forecast values.

For validation purposes, the prediction model has been successfully deployed on a real Smart GW, based on a Raspberry Pi 3 Model B (RPi3), receiving air temperature data from a SN (in detail, a LoPy4 board [23] equipped with a Si7006-A20 temperature sensor [24]) installed in a greenhouse and forwarding these data to the Cloud. When one or more

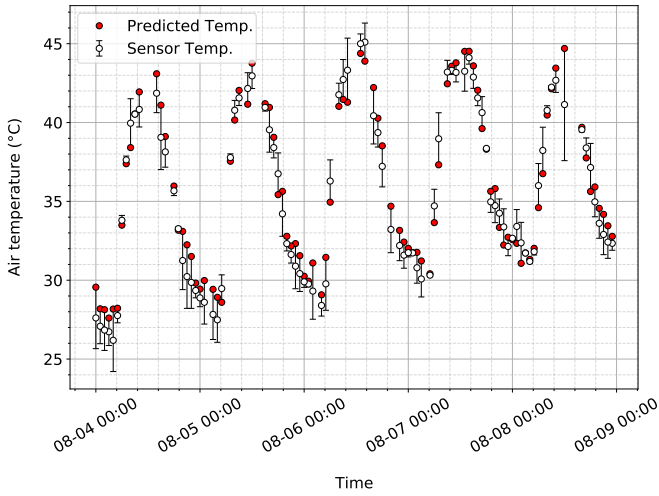


Fig. 2. Predicted air temperatures compared with data collected from sensors during a 5-day period.

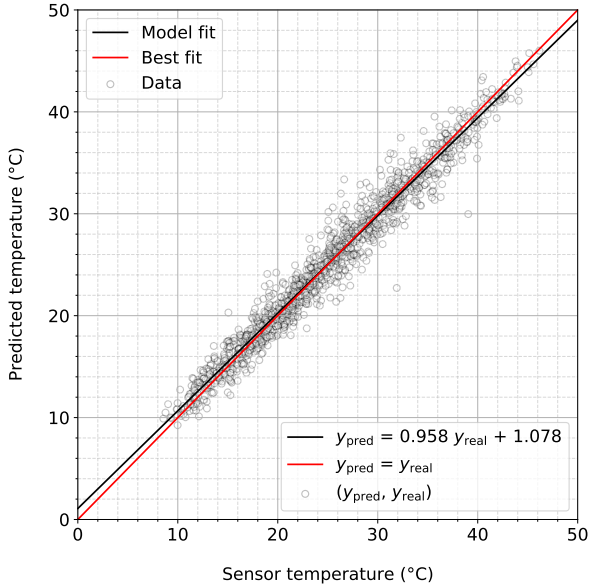


Fig. 3. Comparison between air temperature data, measured by sensors, and predicted values (respectively,  $y_{\text{real}}$  and  $y_{\text{pred}}$ ) on the test set.

sensor data—which are collected with a sampling interval of 10 min—are lost, weather data are retrieved by the Smart GW from the DarkSky repository and prepared as inputs for the ANN-based prediction algorithm which, in turn, estimates the values of the lost data and forwards them to the Cloud. Moreover, the estimated value can be locally exploited to control the opening or closing of the greenhouse’s rooftop, avoiding the need of another entity, at Cloud level, taking the decision and communicating it to the GW. This suggests that a greenhouse could be completely locally managed by a Smart GW, thus reducing network traffic and latency, as the decision is not taken in the Cloud.

## V. CONCLUSIONS

In this paper, a possible approach to embed intelligence into a Smart GW, acting as data collector for SNs measuring air temperature inside a greenhouse, has been proposed. In particular, we have developed an ANN-based prediction model to locally forecast the greenhouse’s air temperature in the presence of sensor data loss. The obtained data can be exploited to regulate air temperature: for example, by controlling actuators installed in the greenhouse.

The external weather conditions of the greenhouse (i.e., apparent temperature, dew point, air humidity and temperature, and UV index) and a time reference (i.e., hour of the day and harvest month), which can be retrieved from locally-deployed sensors or from the Cloud (i.e., from the DarkSky weather data repository), have been adopted as input variables for the ANN-based model, which predicts the air temperature inside the greenhouse with a RMSE of  $1.50^\circ\text{C}$ , a MAPE of 4.91%, and a  $R^2$  of 0.965. Although the ANN-based prediction model, which is composed by only 4 hidden layers and 1018 parameters, has been especially designed to be lightweight and executable by a constrained IoT device (namely, a Smart GW built on a RPi3), it can also be executed in the Cloud to forecast future air temperature values, which can help the farmer in managing his/her greenhouse.

In the future, other ML-based architectures will be evaluated and compared with the one selected in this paper, in order to predict the greenhouse’s air temperature. Moreover, other models will be deployed, in order to forecast other relevant internal variables for the greenhouse (e.g., air humidity).

## ACKNOWLEDGEMENTS

This work has received funding from the European Union’s Horizon 2020 research and innovation program ECSEL Joint Undertaking (JU) under Grant Agreement No. 783221, AFar-Cloud project - “Aggregate Farming in the Cloud.” The JU receives support from the European Union’s Horizon 2020 research and innovation programme and the nations involved in the mentioned projects. The work reflects only the authors’ views; the European Commission is not responsible for any use that may be made of the information it contains. The work of Luca Davoli is partially funded by the University of Parma, under “Iniziativa di Sostegno alla Ricerca di Ateneo” program, “Multi-interface IoT sYstems for Multi-layer Information Processing” (MIoTYMIP) project. The work of Gaia Codeluppi is partially funded by the Regione Emilia Romagna, under “Sistemi IoT per la raccolta e l’elaborazione dei dati efficienti in agricoltura di precisione e sostenibile” (AgrIoT) Ph.D. scholarship.

## REFERENCES

- [1] A. Kumar, G. N. Tiwari, S. Kumar, and M. Pandey, “Role of Greenhouse Technology in Agricultural Engineering,” *International Journal of Agricultural Research*, vol. 5, no. 9, pp. 779–787, 2010, doi:10.3923/ijar.2006.364.372.
- [2] G. Bot, “Physical Modeling of Greenhouse Climate,” *IFAC Proceedings Volumes*, vol. 24, no. 11, pp. 7–12, 1991, doi:10.1016/B978-0-08-041273-3.50006-9.

- [3] D. Shinde and N. Siddiqui, "IoT Based Environment change Monitoring Controlling in Greenhouse using WSN," in *2018 International Conference on Information, Communication, Engineering and Technology (ICICET)*, 2018, pp. 1–5, doi:10.1109/ICICET.2018.8533808.
- [4] G. Codeluppi, A. Cilfone, L. Davoli, and G. Ferrari, "VegIoT Garden: a modular IoT Management Platform for Urban Vegetable Gardens," in *2019 IEEE International Workshop on Metrology for Agriculture and Forestry (MetroAgriFor)*, Portici, Italy, 2019, pp. 121–126, doi:10.1109/MetroAgriFor.2019.8909228.
- [5] T. Moon, S. Hong, H. Young Choi, D. Ho Jung, S. Hong Chang, and J. Eek Son, "Interpolation of Greenhouse Environment Data using Multilayer Perceptron," *Computers and Electronics in Agriculture*, vol. 166, p. 105023, 2019, doi:10.1016/j.compag.2019.105023.
- [6] M. Taki, S. Abdanan Mehdizadeh, A. Rohani, M. Rahnama, and M. Rahmati-Joneidabad, "Applied machine learning in greenhouse simulation; new application and analysis," *Information Processing in Agriculture*, vol. 5, no. 2, pp. 253–268, 2018, doi:10.1016/j.inpa.2018.01.003.
- [7] Y. Yue, J. Quan, H. Zhao, and H. Wang, "The Prediction of Greenhouse Temperature and Humidity Based on LM-RBF Network," in *2018 IEEE International Conference on Mechatronics and Automation (ICMA)*, Changchun, China, 2018, pp. 1537–1541, doi:10.1109/ICMA.2018.8484456.
- [8] Y. Lee, P. Tsung, and M. Wu, "Technology Trend of Edge AI," in *2018 International Symposium on VLSI Design, Automation and Test (VLSI-DAT)*, Hsinchu, Taiwan, 2018, pp. 1–2, doi:10.1109/VLSI-DAT.2018.8373244.
- [9] M. Mohammadi, A. Al-Fuqaha, S. Sorour, and M. Guizani, "Deep Learning for IoT Big Data and Streaming Analytics: A Survey," *IEEE Communications Surveys Tutorials*, vol. 20, no. 4, pp. 2923–2960, 2018, doi:10.1109/COMST.2018.2844341.
- [10] A. Kochhar and N. Kumar, "Wireless sensor networks for greenhouses: An end-to-end review," *Computers and Electronics in Agriculture*, vol. 163, p. 104877, 2019, doi:10.1016/j.compag.2019.104877.
- [11] M. Abbasi, M. H. Yaghmaee, and F. Rahnama, "Internet of things in agriculture: A survey," in *2019 3rd International Conference on Internet of Things and Applications (IoT)*, Isfahan, Iran, 2019, pp. 1–12, doi:10.1109/IICITTA.2019.8808839.
- [12] Z. Tafa, F. Ramadani, and B. Cakolli, "The Design of a ZigBee-Based Greenhouse Monitoring System," in *2018 7th Mediterranean Conference on Embedded Computing (MECO)*, Budva, Montenegro, 2018, pp. 1–4, doi:10.1109/MECO.2018.8405966.
- [13] M. T. W. Wiboonjaroen and T. Sooknuan, "The Implementation of PI Controller for Evaporative Cooling System in Controlled Environment Greenhouse," in *2017 17th International Conference on Control, Automation and Systems (ICCAS)*, Jeju, South Korea, 2017, pp. 852–855, doi:10.23919/ICCAS.2017.8204344.
- [14] Z. Zou, Y. Bie, and M. Zhou, "Design of an Intelligent Control System for Greenhouse," in *2018 2nd IEEE Advanced Information Management, Communicates, Electronic and Automation Control Conference (IMCEC)*, Xi'an, China, 2018, pp. 1–1635, doi:10.1109/IMCEC.2018.8469309.
- [15] W. Hongkang, L. Li, W. Yong, M. Fanjia, W. Haihua, and N. Sigrimis, "Recurrent Neural Network Model for Prediction of Microclimate in Solar Greenhouse," *6th IFAC Conference on Bio-Robotics BIORBOTICS 2018*, vol. 51, no. 17, pp. 790–795, 2018, doi:10.1016/j.ifacol.2018.08.099.
- [16] D.-H. Jung, H. Seok Kim, C. Jhin, H.-J. Kim, and S. Hyun Park, "Time-series analysis of deep neural network models for prediction of climatic conditions inside a greenhouse," *Computers and Electronics in Agriculture*, vol. 173, p. 105402, 2020, doi:10.1016/j.compag.2020.105402.
- [17] T. Choudhary, M. Vipul, G. Anurag, and S. Jagannathan, "A comprehensive survey on model compression and acceleration," *Artificial Intelligence Review*, 2020, doi:10.1007/s10462-020-09816-7.
- [18] M. Taki, Y. Ajabshirchi, S. F. Ranjbar, A. Rohani, and M. Matloobi, "Heat transfer and MLP neural network models to predict inside environment variables and energy lost in a semi-solar greenhouse," *Energy and Buildings*, vol. 110, pp. 314–329, 2016, doi:10.1016/j.enbuild.2015.11.010.
- [19] "DarkSky Open Weather Data Repository," Accessed on 4 October 2020. [Online]. Available: <https://darksky.net/dev>
- [20] G. Codeluppi, A. Cilfone, L. Davoli, and G. Ferrari, "LoRaFarM: A LoRaWAN-Based Smart Farming Modular IoT Architecture," *Sensors*, vol. 20, no. 7, 2020, doi:10.3390/s20072028.
- [21] "Podere Campàz - Official Website," Accessed on 5 October 2020. [Online]. Available: <https://www.poderecampaz.com/>
- [22] M. Franzese and A. Iuliano, "Correlation Analysis," in *Encyclopedia of Bioinformatics and Computational Biology*, S. Ranganathan, M. Gribskov, K. Nakai, and C. Schonbach, Eds. Oxford: Academic Press, 2019, pp. 706–721, doi:10.1016/B978-0-12-809633-8.20358-0.
- [23] "LoPy Datasheet v1.0," Accessed on 4 October 2020. [Online]. Available: <https://cdn.sparkfun.com/assets/e/b/2/9/0/lopy4-specsheet.pdf>
- [24] "Si7006-A20 I<sup>2</sup>C Sensor," Accessed on 3 October 2020. [Online]. Available: <https://www.silabs.com/documents/public/datasheets/Si7006-A20.pdf>