A Multi-Model Deep Learning Approach To Address Prediction Imbalances In Smart Greenhouses

Juan Morales-García[a,*], Fernando Terroso-Sáenz[b], José M. Cecilia[c]

[a]*Department of Computer Science, Universidad Católica de Murcia (UCAM), Murcia, Spain*
[b]*Department of Information and Communication Technologies, Universidad Politécnica de Cartagena (UPCT), Murcia, Spain*
[c]*Department of Computer Engineering (DISCA), Universitat Politècnica de València (UPV), Valencia, Spain*

Abstract

The creation of smart greenhouses is playing a crucial role in paving the way toward precision agriculture characterized by enhanced efficiency. Integral to these greenhouses are decision-support systems that leverage sophisticated forecasting algorithms to predict a range of parameters. However, these predictors often employ a single model approach for forecasting all variables of interest, leading to imbalanced predictions where some variables are accurately predicted while others do not. Such inconsistencies can undermine the overall reliability of the decision-support systems. Addressing this challenge, this paper proposes an approach that harnesses the potential of multiple deep-learning models operating concurrently to predict a broad array of environmental parameters within a smart and operational greenhouse. Each model is specifically tailored to concentrate on a distinct subset of target variables, thus ensuring that the overall accuracy of the prediction is optimized. The effectiveness of this approach has been evaluated in a real-world greenhouse setting. The results indicate a substantial improvement, exhibiting more than an 8% enhancement in the Mean Absolute Percentage Error (MAPE) compared to a single-model alternative, particularly in predicting specific environmental variables, confirming the potential for more reliable and precise agricultural decision-support systems.

Keywords: Smart Greenhouses, AIoT, Forecasting, Multi-model Deep Learning, Precision Agriculture

1. Introduction

One of the major challenges of the human race in this modern era is the frequent occurrence of extreme weather events in many parts of the globe (Schmitt et al., 2022). At the same time, the worldwide population would be close to 10 billion people by 2050 (Sadigov et al., 2022). This will generate a tremendous increment in food demand shortly. This situation calls for developing an agriculture industry that is more efficient and precise. In this context, recent years have witnessed a large deployment of innovative technologies to face such an important challenge.

One of these prominent technologies has been the combination of Artificial Intelligence (AI) techniques and the Internet of Things (IoT) for the development of *Smart Greenhouses* able to monitor and control crop thermodynamics in real time to face sudden weather changes that occur mainly in semi-arid climates Ardiansah et al. (2020). Given

*Principal corresponding author
Email addresses: jmorales8@ucam.edu (Juan Morales-García), fernando.terroso@upct.es (Fernando Terroso-Sáenz), jmcecilia@disca.upv.es (José M. Cecilia)

the fact that greenhouses are quite susceptible to the environmental conditions, an important course of action has focused on applying different forecasting algorithms to anticipate future climatic conditions in a target crop area so that the Decision-Support System (DSS) can take appropriate proactive actions based on such estimations Nakhaei et al. (2023); Maraveas (2022).

To compose such predictive models, it is possible to find in the literature a large number of solutions able to predict a single climatic variable (e.g., the inner temperature of a smart greenhouse) Codeluppi et al. (2020); Oh et al. or multiple ones Jin et al. (2021); Tsai et al. (2020). For that goal, different algorithms have been applied from the statistical Sun et al. (2019); Sharma et al. (2022), machine learning Tsai et al. (2020); Kaneda and Mineno (2016) and deep learning fields Guillén-Navarro et al. (2020); Liu et al. (2023). In that sense, most of these approaches follow a *single-method* approach where a unique predictor forecasts all the target variables.

However, the latent patterns and time-related features of each environmental variable might be quite different among them. Hence, the aforementioned *single-model* approaches might provide *imbalanced* results where certain variables are accurately predicted, and others are more poorly anticipated. This might lead to the malfunctioning of a smart greenhouse's DSS as most of its proactive rules usually take into account multiple climatic factors like soil temperature, soil water potential, rainfall, wind speed, temperature, humidity or global radiation Zhai et al. (2020). As an illustrative example, the DSS proposed in Tay et al. (2021) includes a set of chained fuzzy rules that consider the internal temperature, humidity, and radiation level of a crop to asses its meteorological risk. This calls for predictive solutions able to anticipate all the ambient factors with a similar accuracy level.

In this context, the present work relies on the rationale that the combination of multiple predictive models to forecast different ambient factors of a greenhouse will eventually lead to a more accurate and balanced prediction. Based on this hypothesis, the main goal of this study is to forecast the ambient conditions of a smart greenhouse using an ensemble of deep learning algorithms. To do so, the solution follows an integrated learning approach by comprising a group of predictors where each one focuses on predicting a subset of ambient factors. To compose such an ensemble, first, a set of candidate algorithms are evaluated independently to predict a set of environmental factors of an intelligent greenhouse under certain conditions. Based on this evaluation, the minimum set of models that provides the most accurate predictions for all atmospheric variables are eventually selected. Using this integrated learning approach, a solution can be provided wherein the optimal predictive model manages all relevant ambient variables. Given the multi-dimensionality of the DSS's rules, a more accurate multivariate prediction will help DSSs take more reliable proactive actions to manage a greenhouse.

The main findings of this study include:

1. A novel integrated learning framework has been established, merging multiple deep-learning models to predict ambient conditions within a smart greenhouse.
2. Through comprehensive evaluation, a subset of algorithms was chosen to form an ensemble, each specializing in the accurate prediction of distinct environmental factors.
3. The selected models within the ensemble were fine-tuned to provide the most precise forecasts for each atmospheric variable, leading to an optimal multivariate predictive model.
4. The application of the integrated learning approach resulted in a more accurate and multivariate prediction

model, which significantly augments the decision-making capabilities of DSSs in managing greenhouse conditions.

5. The improved prediction accuracy facilitates DSSs to take proactive and reliable actions, ensuring better climate control within the smart greenhouse.
6. The study advances the field of smart agriculture by providing a sophisticated approach to managing the complex dynamics of greenhouse environments, potentially increasing yield and resource efficiency.

The remainder of the paper is organized as follows. Section 2 summarizes state-of-the-art related studies regarding ambient condition forecasting in smart greenhouses. Section 3 describes the use-case setting, and the main features of the target data along with the proposed *multi-model* approach. Section 4 shows the results, analysis, and discussion of such an approach. Section 5 highlights the conclusions and directions for future works

2. Related Works

In this context of precision agriculture in smart greenhouses, there are univariate studies to predict, generally, the temperature inside the greenhouse. In Ruiz et al. (2022) a work that implements different time series libraries for the prediction of the interior temperature of a greenhouse can be observed. In Morales-García et al. (2023) the authors perform a data-driven evaluation from different ML models to forecast the indoor temperature of the greenhouse. Another work aimed at predicting the inside temperature of the greenhouse is shown in Eraliev and Lee (2023), in which the authors perform a performance analysis of various DL models at different time intervals.

Due to the complexity of the greenhouse, it is not enough to develop univariate models, but it is necessary to consider the greenhouse as a whole, in a holistic way, being necessary to develop multivariate AI models. In this context, one of the most widely used models to make predictions within the greenhouse is the LSTM (Long Short-Term Memory). In Ali and Hassanein (2020) a multivariate LSTM is developed to forecast environmental conditions (air temperature, relative humidity, pressure, wind, and dew point) inside a greenhouse. In Patrizi et al. (2022) the authors propose the creation of a virtual soil moisture sensor based on the development of a multivariate LSTM that collects data from another physical sensor. In Jung et al. (2022) another LSTM is also developed to forecast evapotranspiration and relative humidity for moisture control in tomato plantations inside greenhouses.

Other DL models can be used to predict potential crop problems in greenhouses, as shown in the work Magalhães et al. (2021), which implements different machine vision-focused DL models (such as SSD and YOLO) for tomato identification and its state of maturation, being able to detect green and reddish tomatoes, including those occluded by leaves. In Moreira et al. (2022) another work that implements SSD and YOLO as DL models to efficiently detect tomatoes and compare those systems with a proposed histogram-based HSV color space model to classify each tomato and determine its ripening stage. Another interesting work is described in Kour et al. (2022) where authors propose a hydroponic cultivation of saffron in an IoT environment. In this manuscript, an architecture based on IoT sensors is presented, which collect data and send them to the cloud via wireless networks to be analyzed and processed, evaluated using the AquaCrop model Mirsafi et al. (2016), validating that this architecture can obtain better results than a natural crop, which would require a lot of time.

Taking into account all the variables that are collected in a greenhouse, it is difficult to obtain a generic model that adapts well to all of them, regardless of other needs such as computational load or energy consumption Marchisio et al. (2019). That is why it is necessary to build an ensemble of AI models that, depending on the characteristics of the variables to be predicted as well as the computational needs or energy consumption, choose the AI model that best fits that scenario. For example, in Fu et al. (2019) a stacking model integrating six ML models improved the ability to predict the photosynthetic capacity of different tobacco genotypes. Another example of this is Chang et al. (2020), where the stacking model was also better than the existing base models in forecasting air pollution. In Jiang et al. (2023) also shows a work in which an ensemble of ML models improves the forecast of its base models in the context of greenhouse gas emissions. In Gong et al. (2021) a study of multivariate ML and DL techniques is proposed to compare them with their proposal of RNN-LSTM + TCN in the context of crop yield in greenhouses.

As can be seen, there are numerous studies on the greenhouse internal environmental conditions forecast. However, these works are intended to model the thermodynamics of the greenhouse. On the contrary, the present work proposes an *ensemble* of the best models to predict a set of environmental variables.

3. Materials and Methods

This section outlines the use-case setting in which the proposed solution has been tested, the description of the target prediction problem along with all deep learning models used and the integrated learning methodology adopted to solve it.

3.1. Use-case setting

The operational greenhouse that this study is targeting, known as *ETIFA*, is depicted in Figure 1. *ETIFA* is a functioning greenhouse owned by NUTRICONTROL[1], a Spanish company focusing on the development of climate control and automatic fertigation technology. *ETIFA* covers a 50 m^2 surface area and it is situated in Murcia, a semi-arid region in the southeast of Spain with a yearly average temperature of about 25 ºC.

[1] https://nutricontrol.com/

Figure 1: ETIFA: NUTRICONTROL operational greenhouse located at Murcia (Spain)

A modular IoT infrastructure for climate control and fertigation system is deployed inside the *ETIFA* greenhouse. This IoT infrastructure is coordinated by the *OPTIMUM*[2] integrated control system, shown in Figure 2.

Figure 2: OPTIMUM: NUTRICONTROL integrated control system installed inside of the greenhouse

OPTIMUM integrated control system is controlled by a CPU-based node called *OPTIMUM Orchestator*, to which all sensors are connected in a modular way. In this setting, *ETIFA* greenhouse has several sensors connected to the *OPTIMUM* system to measure different environmental variables. Let us call $\mathscr{V}$ the set comprising all such variables, namely,

- **Indoor temperature (IT),** temperature inside the greenhouse measured by a specific thermometer for this purpose. It is an internal variable of the greenhouse. Its range of values is [9.573 - 33.724]$^{\circ}C$.
- **Wet bulb temperature (WBT),** humid interior temperature of the greenhouse collected using a specific thermometer for this purpose. It is an internal variable of the greenhouse. Its range of values is [8.323 - 29.678]$^{\circ}C$.

[2]https://nutricontrol.com/es/controladores-de-riego/control-integrado-optimum/

- **Relative humidity (RH),** percentage of humidity inside the greenhouse. It is an internal variable of the greenhouse. Its range of values is: [53.257 - 100.000]%.

- **Water deficit (WaD),** lack of irrigation of crops in the greenhouse. It is an internal variable of the greenhouse. Its range of values is [0.000 - 10.611]gr/m^3.

- **External temperature (ET),** temperature outside the greenhouse, measured in ºC. It is an external variable to the greenhouse. Its range of values is: [2.465 - 32.689]$^{\circ}C$.

- **External humidity (EH),** humidity outside the greenhouse. It is an external variable to the greenhouse. Its range of values is: [11.398 - 100.000] %.

- **Wind direction (WiD),** wind direction outside the greenhouse. It is an external variable to the greenhouse. Its range of values is: [46.357 - 296.571]$^{\circ}$.

- **Carbon Dioxide (CO_2),** CO_2 reading inside the greenhouse. It is an internal variable of the greenhouse. Its range of values is [363.500 - 408.933]Ppm.

3.2. Dataset

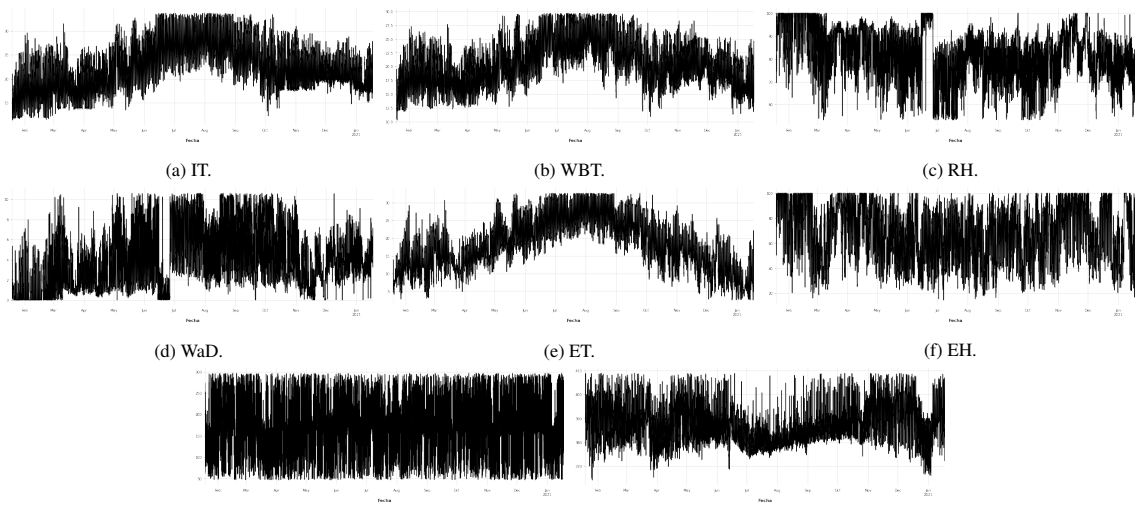
Given the IoT infrastructure described above, three datasets were collected; i.e. $\mathscr{T}_{15}$, $\mathscr{T}_{30}$ and $\mathscr{T}_{60}$, where the measurements of each sensor are reported at different time scales, namely 15, 30 and 60 minutes and comprising the nine target variables described in section 3.1. In that sense, $\mathscr{T}_{30}$ and $\mathscr{T}_{60}$ are aggregations of $\mathscr{T}_{15}$. For the sake of completeness, Table 1 shows the start and end dates of the data, as well as the total number of values for each of the datasets, Table 2 the statistical values and Figure 3 the time-series of each variable in $\mathscr{T}_{15}$.

Name	Start date	End date	# Instances
$\mathscr{T}_{15}$	2018-12-18	2021-01-17	73,103
$\mathscr{T}_{30}$			36,552
$\mathscr{T}_{60}$			18,276

Table 1: Dataset description.

	IT	WBT	RH	WaD	ET	EH	WiD	CO_2
Mean value	21.750	19.430	81.554	3.831	17.912	66.754	172.994	386.717
Std. deviation	5.042	4.358	11.702	2.773	7.440	23.190	54.041	8.631
Min. value	9.573	8.323	53.257	0.000	2.465	11.398	46.357	363.500
Q1	17.950	15.985	73.570	1.518	12.237	47.000	138.000	380.231
Q2	21.283	19.121	81.550	3.500	17.645	67.780	171.778	385.778
Q3	25.174	22.437	90.680	5.666	23.415	87.515	207.174	392.500
Max. value	33.724	29.678	100.000	10.611	32.689	100.000	296.571	408.933

Table 2: Statistical descriptors of the environmental factors of the greenhouse under consideration from the dataset $\mathscr{T}_{15}$. Q{1,2,3} refers to the first, second, and third quartiles.



(a) IT. (b) WBT. (c) RH.

(d) WaD. (e) ET. (f) EH.

(g) WiD. (h) CO_2.

Figure 3: Time-series of each variable in the $\mathscr{T}_{15}$.

Furthermore, Table 3 shows the Pearson Correlation between each of the previously mentioned variables in $\mathscr{T}_{15}$. As can be seen, all variables related to climatic conditions are highly correlated. An example of this is the high correlation of the Internal Temperature (IT) with the Wet Bulb Temperature (WBT) (0.949) and the External Temperature (ET) (0.859) respectively, as they are all temperature-related variables. In addition, all the variables related to irrigation are also related. For instance, the Water deficit (WaD) has a 0.703 Pearson correlation index with IT, -0.919 with Relative Humidity (RH), and -0.655 with External humidity (EH). Therefore, the multivariate nature of this study is justified, to take advantage of all the existing correlations between the variables, so that the AI models can be able to learn better and thus make more accurate predictions.

	IT	WBT	RH	WaD	ET	EH	WiD	CO_2
IT	1.000	**0.949**	-0.482	0.703	0.859	-0.653	-0.042	-0.437
WBT	**0.949**	1.000	-0.241	0.508	0.848	-0.535	-0.045	-0.397
RH	-0.482	-0.241	1.000	**-0.919**	-0.339	0.567	-0.005	0.274
WaD	0.703	0.508	**-0.919**	1.000	0.574	-0.655	-0.008	-0.347
ET	**0.859**	0.848	-0.339	0.574	1.000	-0.631	-0.054	-0.483
EH	-0.653	-0.535	0.567	**-0.655**	-0.631	1.000	0.046	0.401
WiD	-0.042	-0.045	-0.005	-0.008	-0.054	0.046	1.000	0.070
CO_2	-0.437	-0.397	0.274	-0.347	**-0.483**	0.401	**0.070**	1.000

Table 3: Pearson correlation matrix of the environmental variables $\mathscr{V}$ of the greenhouse under consideration. The larger correlation scores in absolute terms for each factor per row are shown in bold.

Lastly, Appendix A shows the autocorrelation and decomposition figures for each of the target variables in $\mathscr{T}_{15}$. As discerned from these plots, every variable is identified as a stationary time series exhibiting daily seasonality, with a cyclical period of 96 lags required to complete one full cycle. The variables exhibit a persistent trend throughout the time series with a modest residual component, notably observed in variables associated with temperature and

irrigation. This stability of the trend contributes to relatively simplified forecasting.

In certain instances, however, the seasonality is contracted to a quarter or expanded to three-quarters of a day. This modification substantially enhances the residual component of the time series, as evidenced by variables concerning humidity, wind, and CO2 concentration. Consequently, these modifications pose considerable challenges for accurate forecasting due to their increased variability.

3.3. Problem formulation

Given the aforementioned dataset, the prediction problem could be formulated as finding a mapping function, $\mathscr{F}$, defined as,

$$\mathscr{F}(\mathscr{V}_{1:t}) \rightarrow \mathscr{V}_{t:t+T}$$

where $\mathscr{V}_{1:t}$ are the historical values of the target variables within the preceding t timestamps and $\mathscr{V}_{t:t+T}$ are their prospective values T timestamps ahead. This formulation sets the stage for exploring various predictive models that can effectively capture and leverage the underlying patterns in the data.

3.4. Deep Learning models

To form the fundamental family of predictors, $\mathscr{P}$, delineated in Section 3.5, the predictive power of five deep learning models has been harnessed. These models have been carefully selected for their distinct strengths and capabilities in handling complex data patterns, hence contributing uniquely to the ensemble model. Their selection is rooted in the premise that each model, with its unique architecture and learning capabilities, can offer a different perspective on the dataset, thereby enriching the overall predictive accuracy and robustness of the ensemble.

A concise description of each model is now provided, elucidating their functionalities and the respective roles they play in the overall architecture. This detailed examination will highlight the unique attributes of each model, from their structural intricacies to the specific types of data patterns they are best equipped to handle. Understanding these characteristics is crucial in comprehending how they collectively contribute to the enhanced predictive capability of the integrated model. Through this exploration, it becomes evident how the synergistic integration of these diverse models can lead to a more robust and accurate forecasting system, capable of adapting to the complexities inherent in the dataset:

1. **MLP**: MultiLayer Perceptron (MLP) is a classic feedforward neural network that mimics the operation of a cluster of biological neurons. Although predominantly used for classification tasks, it is adaptable for regression problems as well. A single perceptron often likened to logistic regression, amalgamates with others to form the multilayer perceptron (MLP), which serves as the structural unit of the Artificial Neural Network (ANN). In the MLP architecture utilized here, three layers are incorporated: an input layer that receives the feature inputs, hidden layers that process these inputs, and an output layer that generates the output. Essentially, each layer endeavors to learn specific weights. The application of activation functions allows ANNs to identify non-linear properties within the network and consequently learn intricate relationships between input and output data.

2. **CNN**: Convolutional Neural Network (CNN) has a wide range of applications across diverse domains, with a primary application in image-based classification tasks. Nevertheless, CNNs can also be deployed for regression problems and adapted to time series data by transforming the information into a format compatible with the convolutional network's inputs. A CNN comprises blocks of filters that facilitate feature extraction from the input via convolutional operations. CNN's advantage over traditional ANNs lies in its automatic learning of filters that identify the most relevant features from the input data.
3. **LSTM**: Long Short-Term Memory (LSTM) is a popular deep learning model due to its ability to incorporate long-term memory, making it especially adept at handling time series data like recurrent models. LSTMs are a subtype of recurrent neural architectures characterized by a state memory and a multilayer cell structure. Each LSTM unit consists of a cell, an input gate, an output gate, and a forget gate. These gates control the flow of data in and out of the cell, allowing the cell to retain values over arbitrary time intervals. Unlike traditional recurrent networks, LSTMs can selectively preserve existing memory instead of overwriting it at each time step, enabling them to detect long-distance dependencies in input sequences.
4. **CNNLSTM**: Convolutional Neural Network + Long Short-Term Memory (CNNLSTM) is a hybrid model that integrates the strengths of both CNN and LSTM models as depicted in Figure 4. Here, the fully connected MLP layer in the convolutional network is replaced with an LSTM network. Consequently, the LSTM assumes the role of accumulating regression results while the CNN focuses on the automated extraction of input features."

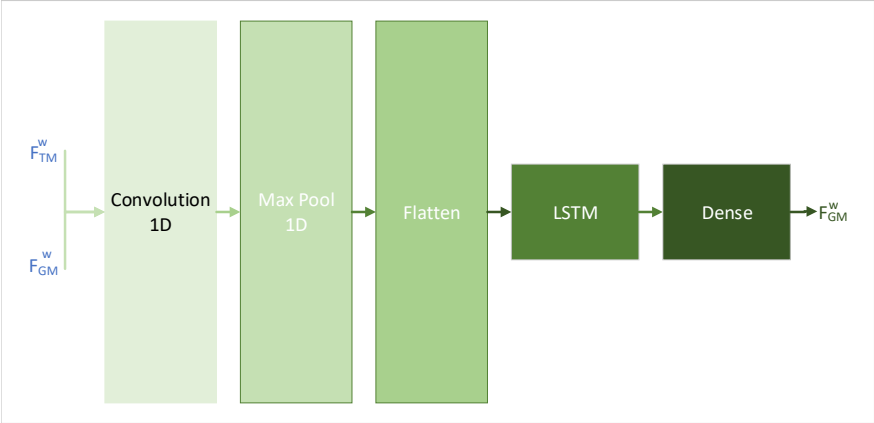


Figure 4: Inner architecture of the CNNLSTM model used in the study.

5. **TFT**: The last model used in this work is the Temporal Fusion Transformer (TFT) proposed in Lim et al. (2021). In that sense, a summary of its inner components is depicted in Figure 5. As can be observed, the system takes as input two types of sequences at a particular time step t.

   First, the multivariate time series $\mathscr{T}_{t-k:t}$ comprises the lags of the target ambient features of the smart greenhouse (sec. 3.2) of the last k time steps. Moreover, the TFT model also accepts known future inputs, $\mathscr{X}_{t+1:t+\tau}$. Given the strong seasonality and autocorrelation of most of the features put forward in sec. 3.2, the month and day associated with each of the target time steps to be predicted were used as future inputs.

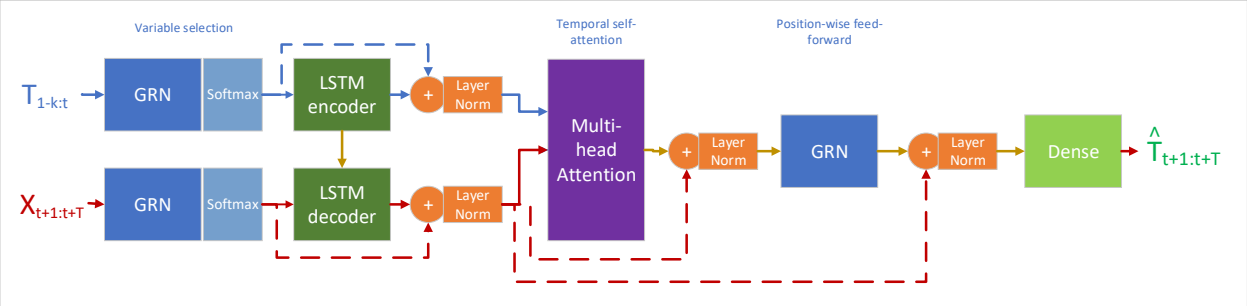


Figure 5: Inner architecture of the Temporal Fusion Transformer used in the study. The residual links are depicted as dashed lines.

Concerning the architecture of the transformer used in the present study, the model first performs a variable selection over the input to assess the relevance and contribution of each variable to the outcome of the other variables. This is quite useful in the present setting due to the correlations among variables described in sec. 3.2. To do so, the model is the Gated Residual Network (GRN), which is one of the building blocks of TFT. In the scope at hand, such a network can be defined as follows,

$$\begin{aligned} GRN_\omega(a) &= LayerNorm(a + GLU_\omega(\ell_1)) \\ \ell_1 &= W_{1,\omega}\ell_2 + b_{1,\omega} \\ \ell_2 &= ELU(W_{2,\omega}a + b_{2,\omega}) \end{aligned} \tag{1}$$

where a is the primary input ($\mathscr{T}_{t-k:t}$ or $\mathscr{X}_{t+1:t+\tau}$) and ELU is the Exponential Linear Unit activation function. As can be observed, this input is processed by a first layer with weight vector $W_{2,\omega}$ and bias term $b_{2,\omega}$ given rise to the embedding ℓ_2. Next, this embedding is processed by an intermediate layer with weight and bias $W_{1,\omega}$ and $b_{1,\omega}$. Finally, the resulting embedding ℓ_1 feeds a Gated Linear Unit (GLU) and is normalized through a standard layer normalization *LayerNorm*. More in detail, the GLU building block comprises two different layers with weights and bias $\langle W_{4,\omega}, b_{4,\omega}\rangle$ and $\langle W_{5,\omega}, b_{5,\omega}\rangle$, so it is defined as follows,

$$GLU_\omega(\gamma) = \sigma(W_{4,\omega}\gamma + b_{4,\omega}) \odot (W_{5,\omega}\gamma + b_{5,\omega})$$

where σ is the sigmoid activation function and $\odot$ is the element-wise Hadamard product. Last, the GRN output is passed through a Softmax layer to compose the final feature weights. Next, an LSTm encoder-decoder is used where the encoder is fed with the lags and the decoder with the known future data to compose uniform temporal features. These features are processed by a Multi-head attention mechanism. Finally, the attention scores are added and normalized with the temporal features and finally passed through a dense layer to compose the prediction outcome $\hat{\mathscr{T}}_{t+1:t+\tau}$.

It is noteworthy that each of these models operates under a distinct architecture, enabling them to identify different patterns within the input sequences. This becomes an integral aspect when devising a framework in which multiple models are orchestrated to function cohesively. To conclude, the set $\mathscr{P}$ includes an instance of each of these architectures, each configured and trained with the parameters delineated in Table 4.

Hyperparameter	Description	MLP	CNN	LSTM	CNNLSTM	TFT
Units	Number of neurons used in hidden layers	70	-	70	-	64
Filters	Features detector	-	64	-	64	-
Kernel size	Filters matrix used to extract the features from the dataset	-	1	-	2	-
Strides	Number of pixels shifts over the input matrix	-	1	-	1	-
Activation function	Function that decides if a neuron should be (or not) activated	Tanh				
Batch size	Size of bach used for training/forecasting	2880				
Epochs	Number of epoch used in training	15000 (+ *EarlyStopping*)				
Optimizer	Function that optimises the learning of the model	Adam				
Loss function	Function used for evaluating the error of the model in each epoch	MSE				
Learning rate	Percentage change with which weights are updated at each iteration	0.003 (+ *ReduceLROnPlateau*)				
Attention heads	Number of attention heads used by the model	-	-	-	-	4
Train-Eval-Test split	Percentage of the dataset for training, validation and testing.	90% train / 10% validation / 10% test				
T	Time horizon of the prediction (in days).	1, 2, 3				

Table 4: A comprehensive breakdown of hyperparameters for each utilized Deep Learning model. The table encapsulates the specific model configuration, illuminating how different hyperparameters are applied across MLP, CNN, LSTM, CNNLSTM, and TFT.

3.5. *Integrated Learning Procedure*

To address the prediction problem as formulated, the present study adopts an integrated learning approach. This approach is illustrated in Figure 6 and comprises two main steps, namely:

1. **Family of Predictors Construction:** The first step entails defining a set of individual predictors, denoted as $\mathscr{P} = \langle p_1, p_2, ..., p_n \rangle$. Each predictor within this set is tasked with processing the historical data, $\mathscr{V}_{1:t}$, to forecast future values $\hat{\mathscr{V}}_{t+T}$. These predictors are designed to operate independently, with the capability to produce predictions, $\hat{\mathscr{V}}^{p}_{t+T}$, based on their respective learned representations and inferential logic.
2. **Optimal Model Combination and Accuracy Maximization:** The subsequent stage involves identifying an optimal combination of models within $\mathscr{P}$ that ensures the global prediction $\hat{\mathscr{V}}_{t+T}$ achieves maximum accuracy. In order to accomplish this, a mapping function $\Theta(\mathscr{P}) \rightarrow \hat{\mathscr{V}}^{\Theta}_{t+T}$ is constructed. Here, each variable $v \in \hat{\mathscr{V}}^{\Theta}_{t+T}$ is essentially predicted by the model $p \in \mathscr{P}$, which demonstrated the highest predictive accuracy for v throughout its training and validation phases. The subset of models utilized by Θ to formulate the prediction $\hat{\mathscr{V}}^{\Theta}_{t+T}$ is denoted as $\mathscr{P}_{multimodel} \subseteq \mathscr{P}$.

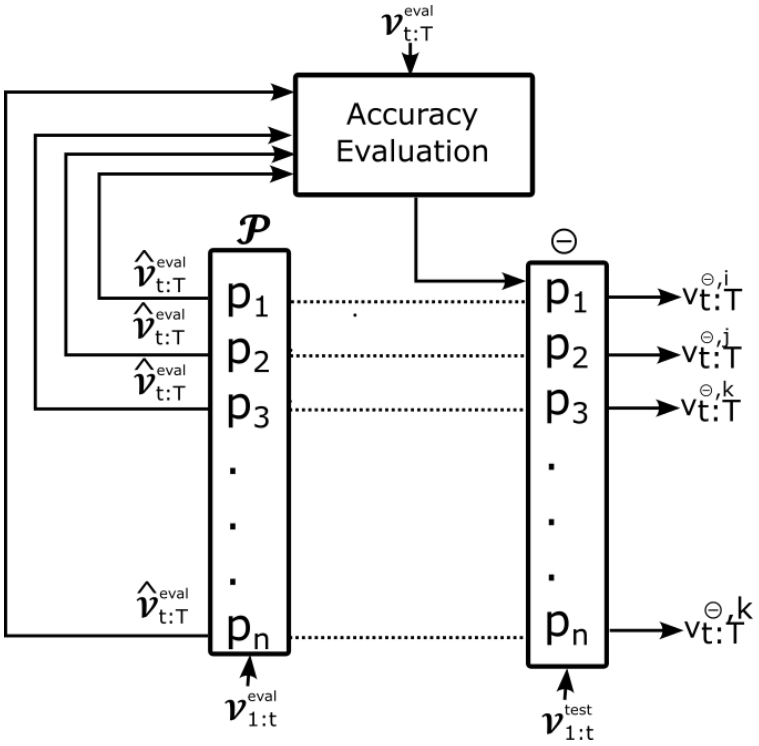


Figure 6: Workflow of the integrated learning followed adopted in this work. In a first stage, each candidate predictor is evaluated in terms of accuracy. Then, a mapping function Θ selects the most accurate predictor to forecast each individual variable in the test set.

Figure 6 encapsulates this two-tiered approach, highlighting the workflow from individual model predictions to the refined, accuracy-focused ensemble output.

4. Evaluation and Discussion

In this section, a comprehensive evaluation and discussion of the methodologies and results are provided. First, the execution environment of all the studied algorithms is described, along with the metrics used to assess their quality. Subsequently, the results from the evaluation of each implemented AI model as a baseline are presented and discussed. Additionally, the forecasts generated by the bespoke deep learning model are presented and analyzed, tailored specifically to the particular scenario of interest.

4.1. Execution environment

To run all the tests presented in this manuscript, a server (named "mercurio") has been utilized, with the following hardware characteristics: An Intel® Xeon® Gold 6226R CPU, with 16 cores at 2.90GHz, 196 GigaBytes DDR4 2933 MHz of RAM memory and 22 MB of cache memory. Two Nvidia® Quadro® RTX 5000 GPUs, with 16GB GDDR6 384 Tensor cores, 3072 CUDA cores and NVLink® PCI Express x16 3.0. A Solid State Drive with 15 TeraBytes.

Regarding the software characteristics, it should be mentioned that "Mercurio" runs on an Ubuntu 20.04 LTS operating system, which has Python version 3.8 with TensorFlow and Keras version 2.12 installed. In addition, for running all the code, the Jupyter development environment was used for the tests, more specifically, the Jupyter Notebooks.

4.2. Evaluation Metrics

For an accurate assessment of the predictive performance of the implemented Deep Learning models, three different metrics are employed: Root Mean Squared Error (RMSE), Mean Absolute Error (MAE), and Mean Absolute Percentage Error (MAPE).

1. **RMSE:** This metric quantifies the dispersion level of the residual values, essentially serving as an indicator of prediction accuracy. It is calculated by using the following formula: $\text{RMSE}(y,\hat{y}) = \sqrt{\frac{\sum_{i=0}^{N-1}(y_i-\hat{y}_i)^2}{N}}$
2. **MAE:** This metric provides the mean deviation between the predicted and actual value at each forecast point, thereby indicating the average magnitude of the prediction errors. It is defined by the formula: $\text{MAE}(y,\hat{y}) = \frac{\sum_{i=0}^{N-1}|y_i-\hat{y}_i|}{N}$
3. **MAPE:** This metric communicates the size of the absolute error in terms of percentage, providing a relative measure of prediction accuracy. It is calculated by using the following formula: $\text{MAPE}(y,\hat{y}) = \frac{100\%}{N}\sum_{i=0}^{N-1}\frac{y_i-\hat{y}_i}{y_i}$

Each of these metrics requires two input parameters: (1) y, the vector containing the ground-truth values; and (2) $\hat{y}$, the vector comprising the values forecasted by the AI models.

4.3. Evaluation of Deep Learning models

The proposed approach was first assessed by calculating the average metric scores of each model in the set $\mathscr{P}$, considering all variables within the set $\mathscr{V}$ across three datasets: $\mathscr{T}15$, $\mathscr{T}30$, and $\mathscr{T}60$. As Table 5 shows, the CNN-LSTM model performs optimally, achieving the lowest scores for RMSE and MAE, and the second-best score for MAPE. Given this outcome, and in a situation where a singular DL model is employed to predict all the greenhouse's target variables, the CNNLSTM model emerges as the most fitting option. Therefore, this model is selected as the uni-model baseline, termed $\mathscr{P}unimodel$, to evaluate the effectiveness of the multi-model approach.

Model	RMSE	MAE	MAPE
CNN	16.157	14.075	57.818
CNNLSTM	**13.764**	**11.581**	30.848
LSTM	16.046	13.393	46.313
MLP	14.905	12.350	43.436
TFT	15.175	12.343	**22.427**

Table 5: Average validation error of the models in $\mathscr{P}$ considering all the variables, datasets and time horizons.

Then, an investigation was conducted to determine if other models in $\mathscr{P}$ could produce a lower validation error than CNNLSTM for each target variable. Table 6 reveals that while the CNNLSTM model predicts the EH, ET, and WiD variables most accurately, the TFT model outperforms the others in predicting the CO_2, IT, RH, and WBT features, and the CNN model performs better for the WaD variable. Interestingly, among the pairs of variables with the highest correlation (Table 3), only the IT-WBT pair is most accurately predicted by the same model. The other pairs of variables, such as RH and WaD (with a correlation score of -0.919), are more accurately predicted by different models - TFT and CNN, respectively.

Therefore, the subset $\mathscr{P}multimodel$ (see Section 3.5) comprises the instances of CNNLSTM, TFT, and CNN models. The mapping function Θ then becomes $\Theta = \langle EH : CNNLSTM,\ ET : CNNLSTM,\ WiD : CNLLST,\ IT :$

$TFTRH : TFTM,\ WBT : TFT, CO_2 : TFT,\ WaD : CNN\rangle$. In this way, the CNNLSTM model is responsible for predicting the EH, ET, and WiD variables, the TFT model for IT, RH, and WBT, and the CNN model for the WaD feature. It is important to note that the CNNLSTM instance included in $\mathscr{P}multimodel$ is the same to the one defined as $\mathscr{P}_{unimodel}$.

Variable	Model	RMSE	MAE	MAPE
$C0_2$	CNNLSTM	8.266	7.043	1.786
	TFT	7.449	5.422	1.384
EH	CNNLSTM	20.093	15.940	33.847
ET	CNNLSTM	3.050	2.688	58.752
IT	CNNLSTM	2.214	1.904	9.782
	TFT	1.878	1.509	7.410
RH	CNNLSTM	8.027	6.504	8.362
	TFT	1.393	1.075	28.666
WBT	CNNLSTM	2.423	2.132	13.237
	TFT	1.894	1.554	9.379
WiD	CNNLSTM	68.779	58.862	35.032
WaD	CNNLSTM	1.390	1.067	57.023
	CNN	1.047	0.737	55.064

Table 6: Comparison between the average evaluation metrics obtained by the CNNLSTM model and the alternative model (if any) that achieved better accuracy for each variable considering the three target datasets and time horizons.

It is worth noting that the findings underscore the value of a multi-model approach in effectively predicting different variables in complex systems such as greenhouses. The evaluation outcomes imply that different DL models, each with their own strengths and expertise, can better cater to the specific nature and requirements of different target variables. Therefore, it is crucial to carefully match the right models to the right variables for optimum predictive performance, as it is shown with the combination of CNNLSTM, TFT, and CNN.

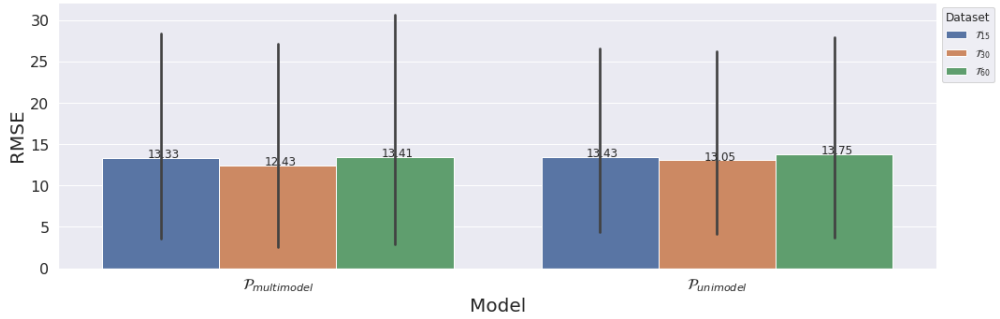
4.4. Evaluation of Multi-model approach

Following the specification of the model subset, $\mathscr{P}multimodel$, and the function Θ, an analysis was conducted to ascertain the performance enhancement of the multi-model strategy. The error metrics of both $\mathscr{P}_{multimodel}$ and $\mathscr{P}_{unimodel}$, shown in Table 7, indicate a similar performance for the variables EH, ET, IT, and WiD, as the CNNLSTM instance was used to generate predictions for these variables in both approaches. However, the multi-model approach performed better than the single-model approach for most of the other variables. Specifically, a significant reduction in MAPE was observed for the RH variable, with a decline of over 4% (16.161 vs 11.952), and an almost 9% reduction for the WaD variable (41.229 vs 30.907). Besides, given this variable the multi-model approach achieved the best accuracy considering the 3 metrics. This also occurred for the RH and CO_2 parameters. However, it is also true that, in the case of the WBT, $\mathscr{P}_{unimodel}$ obtained slight better RMSE and MAE results. Last, in global terms, the last two rows of Table 7 shows that the average MAPE reduction across all variables was approximately 2% (from 24.011 to 22.012), the relative MAE reduction was 8% (12.272 vs 11.288) and 7.7% (14.852 vs 13.716) in the case of the RMSE. This showed a consistent and global improvement of the integrated learning approach with respect the unimodel alternative.

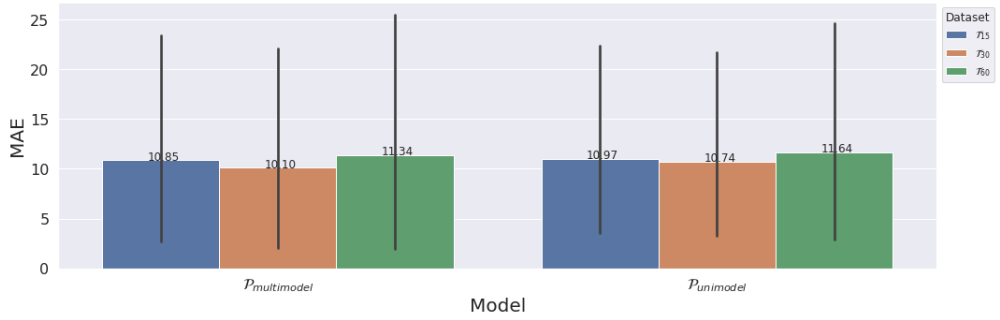
Variable	Model name	RMSE	MAE	MAPE
CO_2	$\mathscr{P}_{unimodel}$	13.066	10.355	2.627
	$\mathscr{P}_{multimodel}$	**7.933**	**5.566**	**1.428**
EH	$\mathscr{P}_{unimodel}$	21.984	18.005	44.455
	$\mathscr{P}_{multimodel}$			
ET	$\mathscr{P}_{unimodel}$	3.381	2.738	32.350
	$\mathscr{P}_{multimodel}$			
IT	$\mathscr{P}_{unimodel}$	1.034	0.848	4.176
	$\mathscr{P}_{multimodel}$			
RH	$\mathscr{P}_{unimodel}$	12.601	10.446	16.161
	$\mathscr{P}_{multimodel}$	**9.007**	**7.802**	**11.952**
WaD	$\mathscr{P}_{unimodel}$	2.383	2.028	41.229
	$\mathscr{P}_{multimodel}$	**1.858**	**1.563**	**30.907**
WBT	$\mathscr{P}_{unimodel}$	**1.890**	**1.607**	10.272
	$\mathscr{P}_{multimodel}$	2.052	1.638	**10.012**
WiD	$\mathscr{P}_{unimodel}$	62.480	52.146	40.818
	$\mathscr{P}_{multimodel}$			
Average	$\mathscr{P}_{unimodel}$	14.852	12.272	24.011
	$\mathscr{P}_{multimodel}$	**13.716**	**11.288**	**22.012**

Table 7: Comparison between the test metrics obtained by a single CNNLSTM instance ($\mathscr{P}_{unimodel}$) and the multi-model approach ($\mathscr{P}_{multimodel}$) for each target variable considering the three datasets and time horizons.

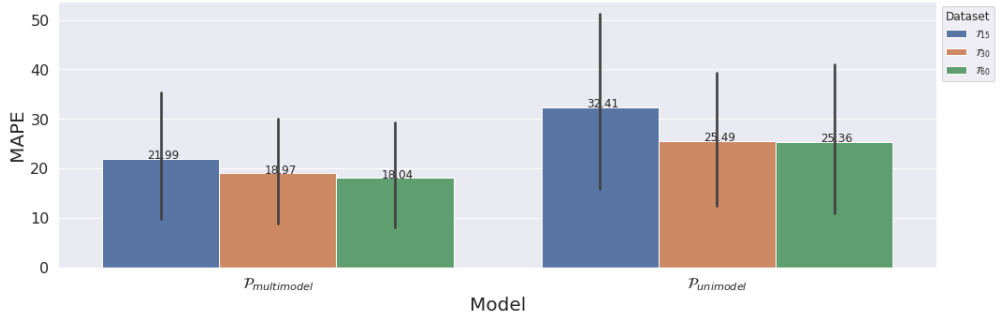
Figures 7, 8, and 9 elucidate the variation in test errors relative to the time horizon T (1, 2, or 3 days) for each of the three datasets $\mathscr{T}_{15}$, $\mathscr{T}_{30}$, and $\mathscr{T}_{60}$. Concerning Fig. 7, where the time horizon T was set to 24 hours, it is interesting to observe that $\mathscr{P}_{multimodel}$ and $\mathscr{P}_{multimodel}$ achieved their best MAE and RMSE results when the time frequency of the dataset was set to 30 minutes ($\mathscr{T}_{30}$) and the higher error scores were achieved when the temporal frequency was set to 60 minutes ($\mathscr{T}_{60}$). This is consistent with the fact that 4 out of 8 variables (EH, ET, IT and WiD) are predicted by the same model in both systems. Consequently, this type of similar patterns in both approaches are not surprising.



(a) RMSE.

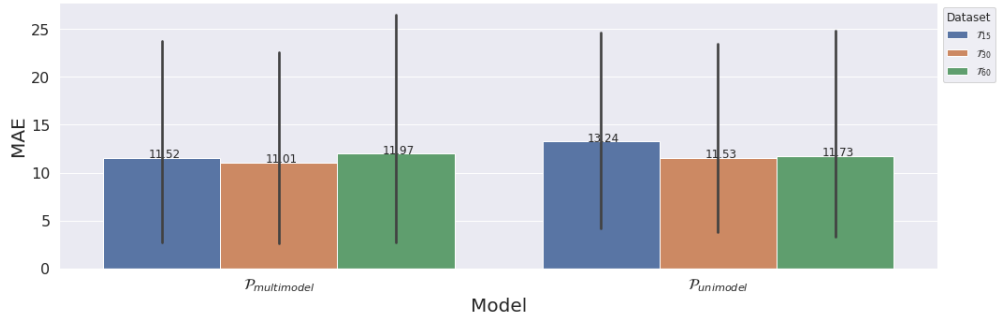


(b) MAE.

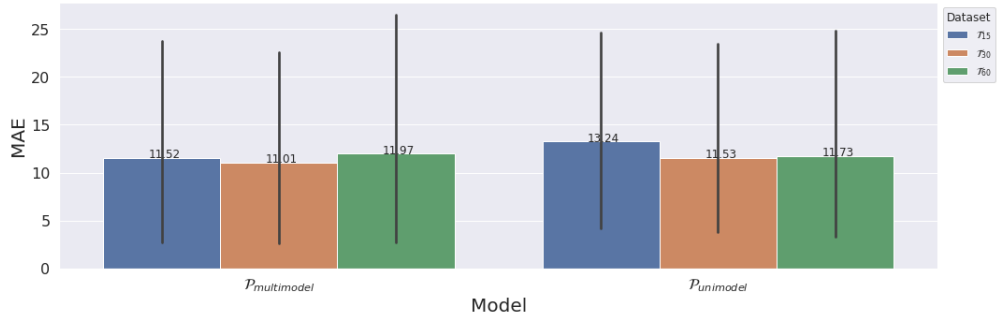


(c) MAPE.

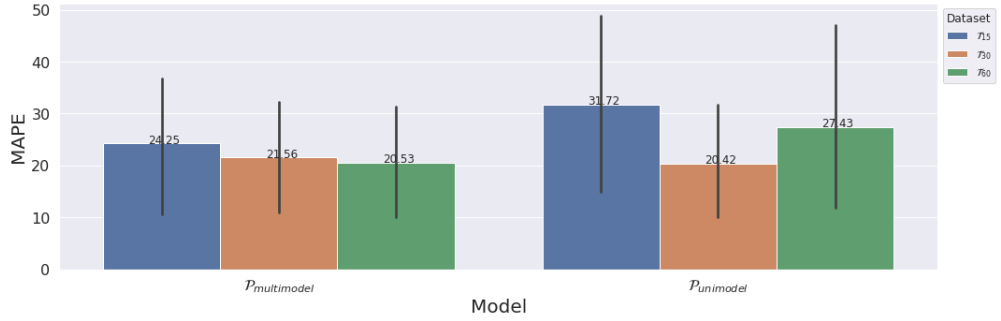
Figure 7: Test scores of the multi-model approach and the CNNLSTM model for T=1 day given the 3 datasets.



(a) RMSE.



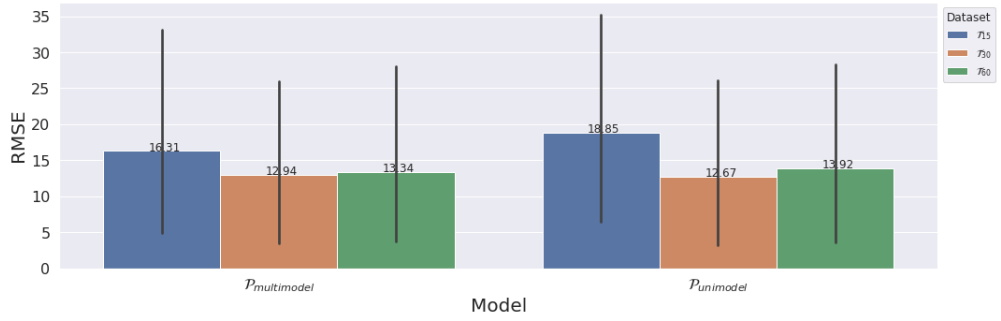
(b) MAE.



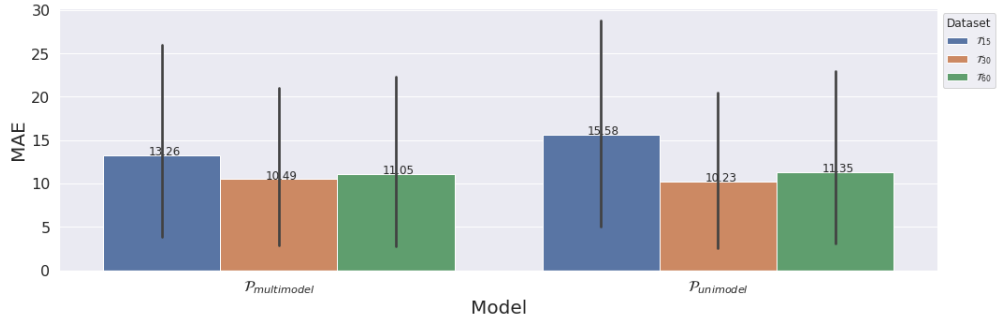
(c) MAPE.

Figure 8: Test scores of the multi-model approach and the CNNLSTM model for T=2 days given the 3 datasets.

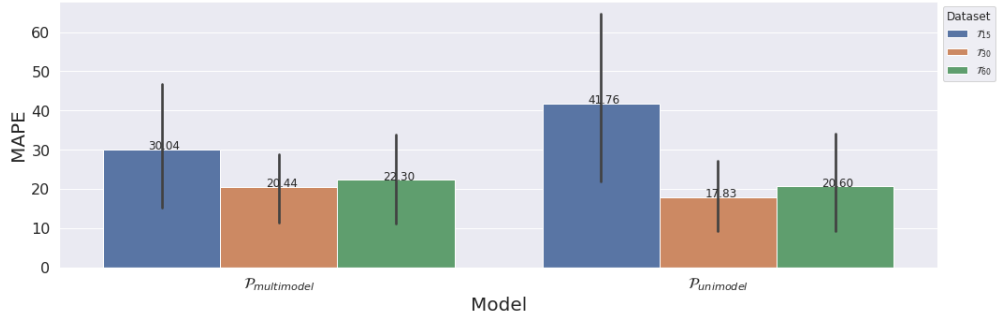
Regarding Fig. 8, where T was set to 48h, it is noteworthy to mention that, in this case, the MAPE score exhibited a steady decrement in the case of $\mathscr{P}_{multimodel}$ as long as the time frequency moved from 15 to 60 minutes as Fig. 8c shows. This behaviour was not observed in $\mathscr{P}_{unimodel}$ where the MAPE was slightly lower for $\mathscr{T}_{30}$ than for the other two datasets. This caused that the MAPE obtained by the integrated learning approach was below 25% for the three datasets under consideration.



(a) RMSE.



(b) MAE.



(c) MAPE.

Figure 9: Test scores of the multi-model approach and the CNNLSTM model for T=3 days given the 3 datasets.

As far as Fig. 9 is concerned, the results when $T = 72h$ show that the degradation of $\mathscr{P}_{multimodel}$ with $\mathscr{T}_{15}$ with respect $\mathscr{T}_{30}$ and $\mathscr{T}_{60}$ was lower than $\mathscr{P}_{unimodel}$. For example, Fig. 9b shows that the MAE increment of $\mathscr{P}_{multimodel}$ when the time frequency moved from 30 to 15 minutes was 20,80% (10.49 vs 13.26) but this increment was much larger for $\mathscr{P}_{unimodel}$, 34,34% (10,23 vs 15,58).

All in all, the effectiveness of the multi-model approach escalated proportionally to the length of the time horizon T, as evidenced by the RMSE and MAE measures. Specifically, the average RMSE reduction was 2.72% (T=1 days, Figure 7), 5.21% (T=2 days, Figure 8), and 6.69% (T=3 days, Figure 9). A similar trend was observed for the MAE score, with average reductions of 3.26%, 5.63%, and 6.78% for T=1, T=2, and T=3 respectively. Meanwhile, the average MAPE reduction was 41.12% for T=1, 20.12% for T=2, and 10.18% for T=3. Nevertheless, it is also possible to observe certain shortcomings of the proposal given the aforementioned evaluation. First, the need to use multiple DL models in parallel poses certain performance requirements that may not be fulfilled by devices installed in fog-computing settings. This may limit the deployment of the proposed solution in certain environments. Moreover,

the actual accuracy improvement of the integrated learning mechanism occurs with large time horizons (T) defined (above 2 days). Consequently, the current solution might not provide an actual improvement concerning an unimodel mechanism when short-term predictions are required.

5. Conclusions and Future Work

Precision agriculture relies on predicting environmental and irrigation variables within smart greenhouses accurately. While various deep learning models have been utilized to tackle this prediction task, they often operate under the assumption of deploying a single model, leading to imbalanced predictions across the different targeted variables.

In the research conducted, this norm was challenged by investigating the efficacy of employing multiple predictors, each focusing on a distinct subset of variables. The objective was to improve the overall prediction accuracy and balance across the target variables. A variety of deep learning architectures were deployed as candidate models, and performed an initial validation analysis. This was carried out to establish a mapping function that determined the optimal model to predict each specific variable.

The resulting multi-model approach demonstrated consistent improvement in prediction accuracy across different time frequencies and horizons. In terms of Mean Absolute Percentage Error (MAPE), reductions of up to 8% were observed compared to a single-model approach used to estimate all target variables. These findings are particularly beneficial, as they promise to supply smart greenhouse recommendation systems with more accurate multivariate inputs, thus enhancing the precision of these systems.

Looking ahead, the practical computational cost of the multi-model approach is intended to be assessed, especially concerning deployment within an Internet of Things (IoT) infrastructure. A key aspect of future work will be examining how this multi-variate approach performs in edge computing devices, allowing us to weigh the improvement in prediction accuracy against any potential increases in energy consumption due to running multiple deep learning models in parallel. This will provide valuable insights into the trade-off between prediction improvement and energy efficiency when applying the multi-model approach in a real-world context. Finally, there are plans to investigate the application of transfer learning and federated learning to enhance the predictive accuracy and efficiency of the presented models in smart agricultural environments, expanding its applications to enhanced leaf disease assessment and classification.

Declarations

Ethical Approval

Not applicable.

Conflict of interest

The authors declare that they have no conflict of interest.

Authors' contributions

Conceptualization, J.M.G.; methodology, J.M.G. and F.T.S; software, J.M.G. and F.T.S; validation, J.M.C., and F.T.S; formal analysis, J.M.C; investigation, J.M.G. and F.T.S.; writing—original draft preparation, J.M.G. and F.T.S.; writing—review and editing, J.M.G., F.T.S. and J.M.C.; visualization, J.M.G. and F.T.S.; supervision, J.M.C.; project administration, J.M.C.; funding acquisition, J.M.G., F.T.S. and J.M.C.

Consent to publish

All authors have read and agreed to the published version of the manuscript.

Funding

This work is derived from R&D projects RTC2019-007159-5, PID2020-112827GB-I00, as well as the Ramon y Cajal Grant RYC2018-025580-I, funded by

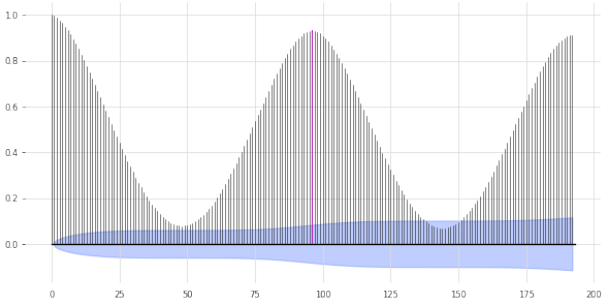
MCIN/AEI/10.13039/501100011033, "FSE invest in your future".

Availability of data and materials

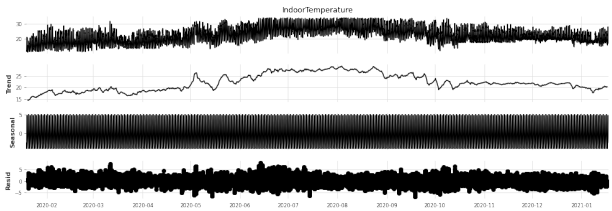
All data and materials are available on request from the authors of this paper.

Appendix A. Exploratory Data Analysis

Figs. A.10 - A.19 below show the autocorrelation function as well as the seasonal decomposition (time series, trend, seasonality, residuals) of each of the variables belonging to the 15-min dataset ($\mathcal{T}_{15}$).

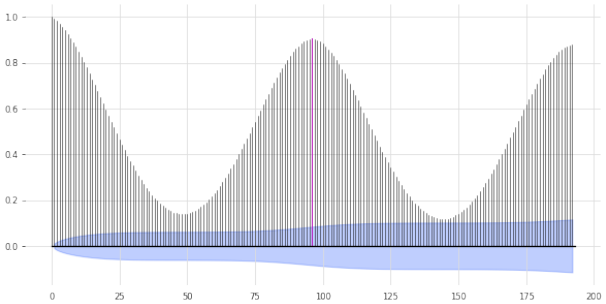


(a) Autocorrelation Function (ACF) of the IT variable.

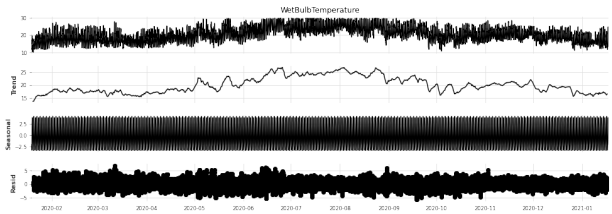


(b) Seasonal decomposition of the IT variable.

Figure A.10: Exploratory Data Analysis (EDA) of the IT variable.

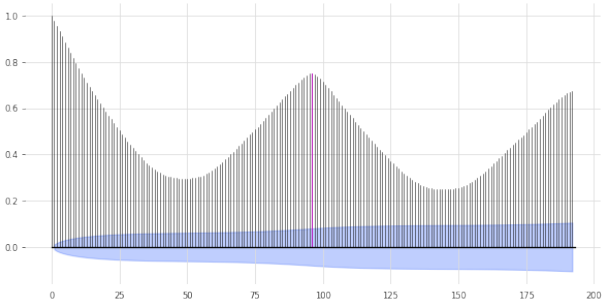


(a) Autocorrelation Function (ACF) of the WBT variable.

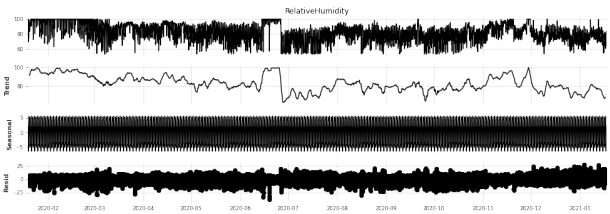


(b) Seasonal decomposition of the WBT variable.

Figure A.11: Exploratory Data Analysis (EDA) of the WBT variable.

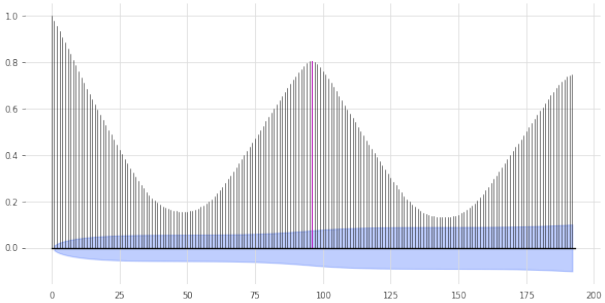


(a) Autocorrelation Function (ACF) of the RH variable.

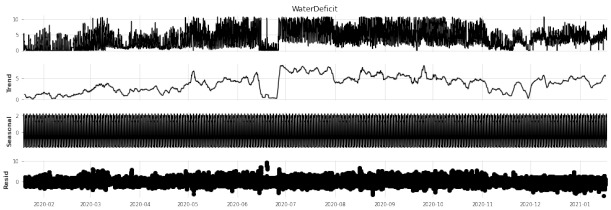


(b) Seasonal decomposition of the RH variable.

Figure A.12: Exploratory Data Analysis (EDA) of the RH variable.

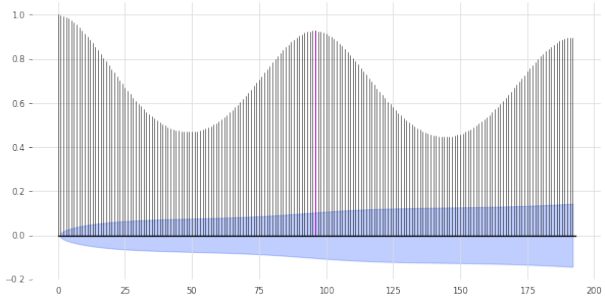


(a) Autocorrelation Function (ACF) of the WaD variable.

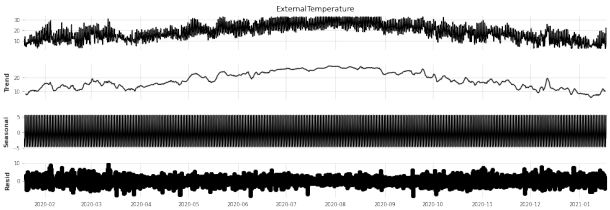


(b) Seasonal decomposition of the WaD variable.

Figure A.13: Exploratory Data Analysis (EDA) of the WaD variable.

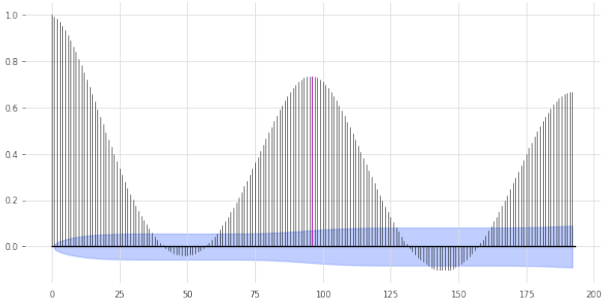


(a) Autocorrelation Function (ACF) of the ET variable.

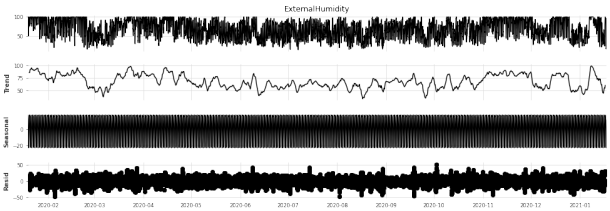


(b) Seasonal decomposition of the ET variable.

Figure A.14: Exploratory Data Analysis (EDA) of the ET variable.

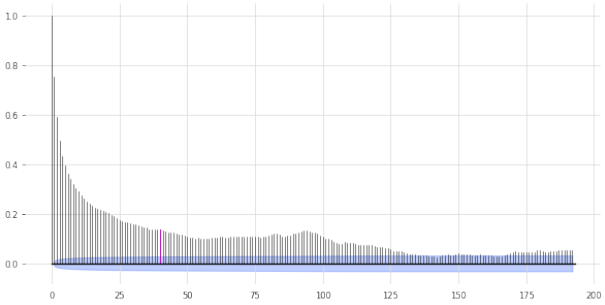


(a) Autocorrelation Function (ACF) of the EH variable.

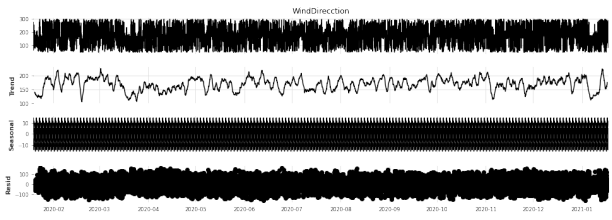


(b) Seasonal decomposition of the EH variable.

Figure A.15: Exploratory Data Analysis (EDA) of the EH variable.

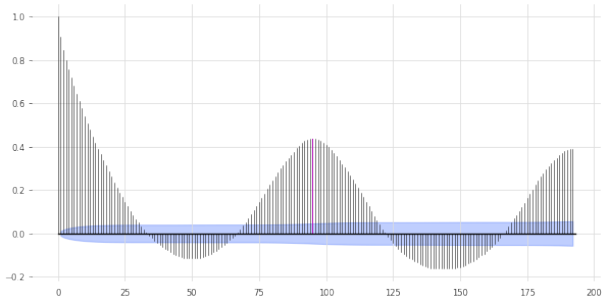


(a) Autocorrelation Function (ACF) of the WiD variable.

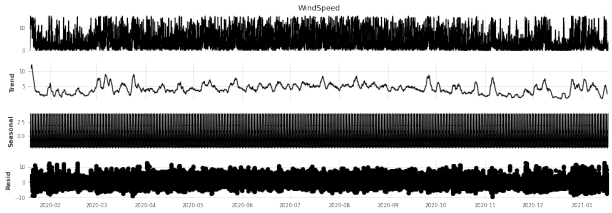


(b) Seasonal decomposition of the WiD variable.

Figure A.16: Exploratory Data Analysis (EDA) of the WiD variable.

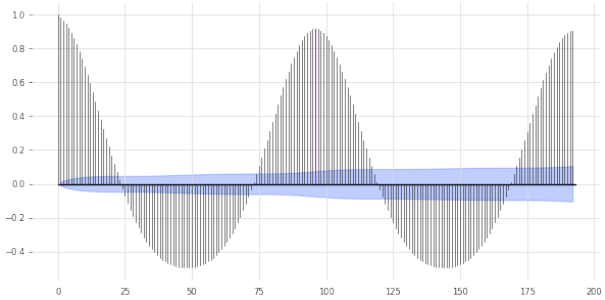


(a) Autocorrelation Function (ACF) of the WS variable.

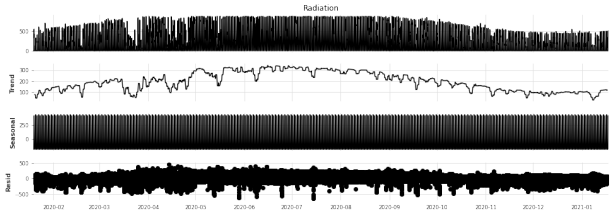


(b) Seasonal decomposition of the WS variable.

Figure A.17: Exploratory Data Analysis (EDA) of the WS variable.

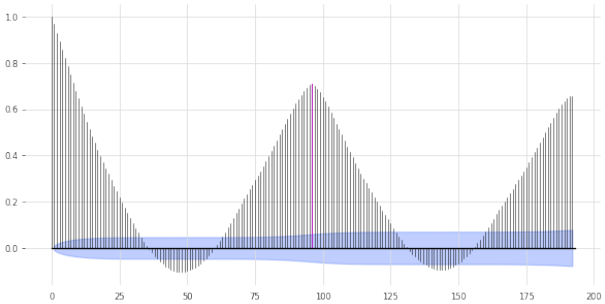


(a) Autocorrelation Function (ACF) of the Rad variable.

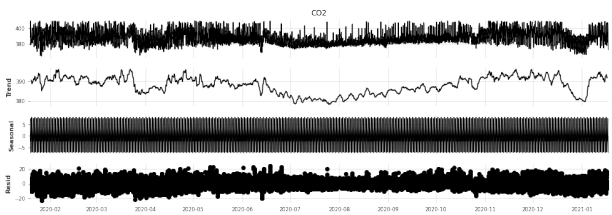


(b) Seasonal decomposition of the Rad variable.

Figure A.18: Exploratory Data Analysis (EDA) of the Rad variable.



(a) Autocorrelation Function (ACF) of the CO_2 variable.



(b) Seasonal decomposition of the CO_2 variable.

Figure A.19: Exploratory Data Analysis (EDA) of the CO_2 variable.

References

Ali, A., Hassanein, H.S., 2020. Time-series prediction for sensing in smart greenhouses, in: GLOBECOM 2020-2020 IEEE Global Communications Conference, IEEE. pp. 1–6.

Ardiansah, I., Bafdal, N., Suryadi, E., Bono, A., 2020. Greenhouse monitoring and automation using arduino: a review on precision farming and internet of things (iot). International Journal on Advanced Science, Engineering and Information Technology 10, 703–709.

Chang, Y.S., Abimannan, S., Chiao, H.T., Lin, C.Y., Huang, Y.P., 2020. An ensemble learning based hybrid model and framework for air pollution forecasting. Environmental Science and Pollution Research 27, 38155–38168.

Codeluppi, G., Cilfone, A., Davoli, L., Ferrari, G., 2020. Ai at the edge: a smart gateway for greenhouse air temperature forecasting, in: 2020 IEEE international workshop on metrology for agriculture and forestry (MetroAgriFor), IEEE. pp. 348–353.

Eraliev, O., Lee, C.H., 2023. Performance analysis of time series deep learning models for climate prediction in indoor hydroponic greenhouses at different time intervals. Plants 12, 2316.

Fu, P., Meacham-Hensold, K., Guan, K., Bernacchi, C.J., 2019. Hyperspectral leaf reflectance as proxy for photosynthetic capacities: An ensemble approach based on multiple machine learning algorithms. Frontiers in Plant Science 10, 730.

Gong, L., Yu, M., Jiang, S., Cutsuridis, V., Pearson, S., 2021. Deep learning based prediction on greenhouse crop yield combined tcn and rnn. Sensors 21, 4537.

Guillén-Navarro, M.A., Martínez-España, R., Bueno-Crespo, A., Morales-García, J., Ayuso, B., Cecilia, J.M., 2020. A decision support system for water optimization in anti-frost techniques by sprinklers. Sensors 20, 7129.

Jiang, Z., Yang, S., Smith, P., Pang, Q., 2023. Ensemble machine learning for modeling greenhouse gas emissions at different time scales from irrigated paddy fields. Field Crops Research 292, 108821.

Jin, X.B., Zheng, W.Z., Kong, J.L., Wang, X.Y., Zuo, M., Zhang, Q.C., Lin, S., 2021. Deep-learning temporal predictor via bidirectional self-attentive encoder–decoder framework for iot-based environmental sensing in intelligent greenhouse. Agriculture 11, 802.

Jung, D.H., Lee, T.S., Kim, K., Park, S.H., 2022. A deep learning model to predict evapotranspiration and relative humidity for moisture control in tomato greenhouses. Agronomy 12, 2169.

Kaneda, Y., Mineno, H., 2016. Sliding window-based support vector regression for predicting micrometeorological data. Expert Systems with Applications 59, 217–225.

Kour, K., Gupta, D., Gupta, K., Anand, D., Elkamchouchi, D.H., Pérez-Oleaga, C.M., Ibrahim, M., Goyal, N., 2022. Monitoring ambient parameters in the iot precision agriculture scenario: An approach to sensor selection and

hydroponic saffron cultivation. Sensors 22. URL: https://www.mdpi.com/1424-8220/22/22/8905, doi:10.3390/s22228905.

Lim, B., Arık, S.Ö., Loeff, N., Pfister, T., 2021. Temporal fusion transformers for interpretable multi-horizon time series forecasting. International Journal of Forecasting 37, 1748–1764.

Liu, J., Shao, M., et al., 2023. The forecast of power consumption and freshwater generation in a solar-assisted seawater greenhouse system using a multi-layer perceptron neural network. Expert Systems with Applications 213, 119289.

Magalhães, S.A., Castro, L., Moreira, G., Dos Santos, F.N., Cunha, M., Dias, J., Moreira, A.P., 2021. Evaluating the single-shot multibox detector and yolo deep learning models for the detection of tomatoes in a greenhouse. Sensors 21, 3569.

Maraveas, C., 2022. Incorporating artificial intelligence technology in smart greenhouses: Current state of the art. Applied Sciences 13, 14.

Marchisio, A., Hanif, M.A., Khalid, F., Plastiras, G., Kyrkou, C., Theocharides, T., Shafique, M., 2019. Deep learning for edge computing: Current trends, cross-layer optimizations, and open research challenges, in: 2019 IEEE Computer Society Annual Symposium on VLSI (ISVLSI), IEEE. pp. 553–559.

Mirsafi, Z.S., Sepaskhah, A.R., Ahmadi, S.H., Kamgar-Haghighi, A.A., 2016. Assessment of aquacrop model for simulating growth and yield of saffron (crocus sativus l.). Scientia Horticulturae 211, 343–351.

Morales-García, J., Bueno-Crespo, A., Martínez-España, R., Cecilia, J.M., 2023. Data-driven evaluation of machine learning models for climate control in operational smart greenhouses. Journal of Ambient Intelligence and Smart Environments , 1–15.

Moreira, G., Magalhães, S.A., Pinho, T., dos Santos, F.N., Cunha, M., 2022. Benchmark of deep learning and a proposed hsv colour space models for the detection and classification of greenhouse tomato. Agronomy 12, 356.

Nakhaei, M., Ahmadi, A., Gheibi, M., Chahkandi, B., Hajiaghaei-Keshteli, M., Behzadian, K., 2023. A smart sustainable decision support system for water management of power plants in water stress regions. Expert Systems with Applications , 120752.

Oh, K.C., Kim, S.J., Park, S.Y., Cho, L., Lee, C.G., Kim, D.H., . Development of greenhouse internal temperature prediction model based on data characteristics using machine learning. Available at SSRN 4329492 .

Patrizi, G., Bartolini, A., Ciani, L., Gallo, V., Sommella, P., Carratù, M., 2022. A virtual soil moisture sensor for smart farming using deep learning. IEEE Transactions on Instrumentation and Measurement 71, 1–11.

Ruiz, S., Morales-García, J., Calafate, C.T., Cano, J.C., Manzoni, P., Cecilia, J.M., 2022. Evaluation of time-series libraries for temperature prediction in smart greenhouses, in: 2022 18th International Conference on Intelligent Environments (IE), IEEE. pp. 1–7.

Sadigov, R., et al., 2022. Rapid growth of the world population and its socioeconomic results. The Scientific World Journal 2022.

Schmitt, J., Offermann, F., Söder, M., Frühauf, C., Finger, R., 2022. Extreme weather events cause significant crop yield losses at the farm level in german agriculture. Food Policy 112, 102359.

Sharma, S., Saxena, A.K., Bansal, M., 2022. Forecasting of ghg (greenhouse gas) emission using (arima) data driven intelligent time series predicting approach, in: 2022 7th International Conference on Communication and Electronics Systems (ICCES), IEEE. pp. 315–322.

Sun, Q., Liao, B., Tao, Q., 2019. Ecological agriculture development and spatial and temporal characteristics of carbon emissions of land use. Applied Ecology & Environmental Research 17.

Tay, A., Lafont, F., Balmat, J.F., Pessel, N., Lhoste-Drouineau, A., 2021. Decision support system for western flower thrips management in roses production. Agricultural Systems 187, 103019.

Tsai, Y.Z., Hsu, K.S., Wu, H.Y., Lin, S.I., Yu, H.L., Huang, K.T., Hu, M.C., Hsu, S.Y., 2020. Application of random forest and icon models combined with weather forecasts to predict soil temperature and water content in a greenhouse. Water 12, 1176.

Zhai, Z., Martínez, J.F., Beltran, V., Martínez, N.L., 2020. Decision support systems for agriculture 4.0: Survey and challenges. Computers and Electronics in Agriculture 170, 105256.