Data-driven evaluation of Machine Learning models for climate control in operational Smart Greenhouses

Juan Morales-García [a,*], Andrés Bueno-Crespo [a], Raquel Martínez-España [b] and José M. Cecilia [c]
[a] *Computer Science Department, Catholic University of Murcia (UCAM), ES, Spain*
E-mails: jmorales8@ucam.edu, abueno@ucam.edu
[b] *Information and Communications Engineering Department, University of Murcia (UM), ES, Spain*
E-mail: raquel.m.e@um.es
[c] *Computer and Systems Informatics Department, Univesitat Politecnica de Valencia (UPV), ES, Spain*
E-mail: jmcecilia@disca.upv.es

Abstract. Nowadays, human overpopulation is stressing our ecosystems in different ways, agriculture being a critical example as different predictions point towards food shortages in the near future. Accordingly, smart farming is becoming key to the optimization of natural resources so that different crops can be grown efficiently, consuming as few resources as possible. In particular, greenhouses have proved to be an effective way of producing a high volume of vegetables/fruits in a reduced space and within a short time span. Hence, optimizing greenhouse functioning results in less water use and nutrient consumption, less energy use, faster growth, and better product quality. In this article, we carry out an in-depth analysis of different machine learning (ML) models to improve climate control in smart greenhouses. As part of the analysis of the techniques we also considered 3 ways of pre-processing the data, as well as 12-hour and 24-hour forecasting. We focus on forecasting the indoor air temperature of an operational smart greenhouse, i.e. assessing the data anomalies that are inherently present in these environments due to the instability of IoT infrastructures. Several ML models are adapted to time series forecasting to provide an overview of these techniques and to find out which one performs better in this particular scenario. Our results show that, after statistically validating the results, the Random Forest Regression technique gives the best overall result with a mean absolute error of less than 1 degree Celsius.

Keywords: Precision Agriculture, Artificial Intelligence, Machine Learning, Temperature Forecasting, Smart Greenhouses

1. Introduction

Global population growth and shortages of natural products are having a transformative effect on agriculture. Precision agriculture (PA) was created as a discipline that seeks to use the latest technologies to achieve higher crop yields with less investment, greater sustainability and improved food safety [1]. PA is a general term that not only works with extensive agriculture, but also applies its novel techniques to different types of crops, including those grown in indoor facilities such as greenhouses. Along these lines, in recent years, new technologies have been introduced in greenhouses that are able to improve yields in a sustainable way [2]. A greenhouse is an agricultural

*Corresponding author. E-mail: jmorales8@ucam.edu.

structure that seeks to extend the production season by providing controlled indoor microclimatic conditions according to the type of crop [3]. Greenhouse agriculture in semi-arid regions, such as the Mediterranean region, has a high productive potential due to its climatic advantages. However, this potential can be further increased if more technological resources are integrated to control its climatic conditioning, as well as to reduce the costs involved with the use of natural resources, such as water or energy [3].

One of the most economically and environmentally relevant factors in greenhouses is climate control [4]. It is an effective tool for maintaining a satisfactory environment inside the greenhouse that meets the requirements of high-yielding crops as well as reducing energy and water consumption. Maintaining ideal conditions inside the greenhouse requires cooling and heating systems that consume between 65 and 85% of the total energy consumed in the greenhouse [5]. This is particularly relevant in semi-arid regions, it being estimated that the cooling energy consumption in the Mediterranean region is about 100,000 kWh/ha per year [6]. Therefore, one of the main objectives when designing smart greenhouses is to optimize the use of their energy resources in order to reduce their carbon footprint, and thus their environmental impact, while also increasing their economic sustainability [7].

A traditional greenhouse is defined as a predominantly metal structure for agricultural cultivation, covered with plastic sheeting. In these conventional greenhouses, they have a wired infrastructure with few resources and many communication problems, which leads to manual monitoring in many cases. These conventional practices require a lot of resources, especially human resources and possible wastage of energy due to the lack of exhaustive climate control. In addition, the lack of comprehensive climate control leads to lower crop yields, [8, 9]. A few years ago, technological progress made it possible to upgrade the status of greenhouses and create smart greenhouses. These greenhouses use all the resources of technology, sensors, communication networks, framework and communication protocols, as well as data storage,analysis and processing, among other things, to obtain a better performance using a lower amount of resources, while being more economically and environmentally sustainable [10, 11]. The advantages of a smart greenhouse over a conventional one include water savings, as it is possible to measure and control the exact amount of water needed by the plants. There is also a reduction in the use of pesticides, which benefits the production of organic crops, as pests can be controlled more efficiently and treated with organic products. Finally, the energy savings are significant, since actuators responsible for heating, humidifiers, shading, etc. are only activated when they are really needed and are disconnected when the optimum climate control is reached, meaning, there is a high precision in the control [12, 13]. Several works can be found in the literature which deal with optimizing climate control in greenhouses. For instance Revathi, Radhakrishnan and Sivakumaran [14] described a classical control system for a greenhouse using a proportional-integral-derivative (PID) control. The main disadvantage of PID-based greenhouses is that they focus on the current state of a single environmental factor, so actions are taken a posteriori. Some ML methods have been proposed to forecast greenhouse internal variables [15, 16]. However, these works are based on the definition of very limited ML techniques, using models with one or very few input variables, usually obtained from publicly available or artificially created databases that have already been curated. Indeed, current smart greenhouses rely on Internet of Things (IoT) infrastructures that are widely used for climate control [17–19]. These IoT deployments are based on several sensors which measure environmental conditions both inside and outside the greenhouse. However, data quality in such climatically aggressive environments is usually not good enough to develop data-based models like ML ones. In this paper, we analyze in depth the use of ML techniques used to forecast the indoor air temperature of an operational greenhouse and deal with its climate control. Several ML methods are designed for time-series regression of this variable, focusing on different data dimensions such as quality, granularity, seasonality and forecast horizon. Main contributions of the paper include the following:

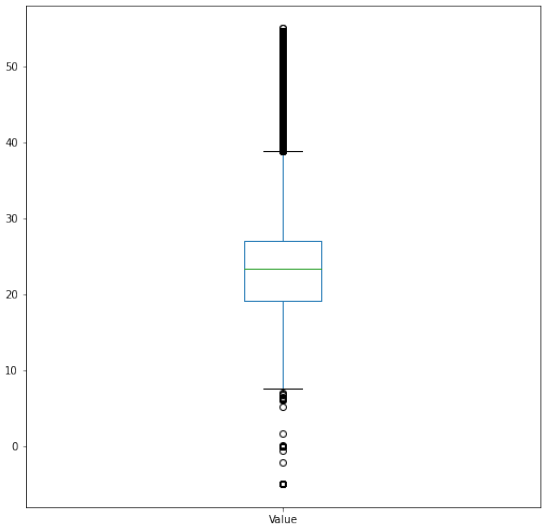
- Design and optimization of up to four ML techniques for indoor temperature forecasting in an operational greenhouse. The techniques designed and adjusted for time series forecasting include the following: Autoregresive (AR), Autoregresive Integrated Moving Average (ARIMA), Random Forest Regressor (RFR) and K-Nearest Neighbors Regressor (KNNR).
- Design and tuning of data pre-processing techniques for data-curation of the raw data obtained from the sensors of an operational greenhouse in real-time.
- Different hourly granularities are considered in the forecasting, as well as a 12-hour and a 24-hour prediction.
- Comparison and analysis of the results obtained. In the analysis, different hourly granularities are considered when making the forecasts, along with forecasts at 12 and 24 hours, and also different types of data sanitization (pre-processing)

The rest of the paper is organised as follows. Section 2 shows related works under the umbrella of ML applied to forecasting climatic variables in greenhouses. Section 3 describes in detail different approaches proposed and the artificial intelligence methods used to compare and evaluate results. Section 4 shows the performance results for the pubsub solution and for the artificial intelligence models. Finally, section 5 presents the main conclusions, and discusses future works.

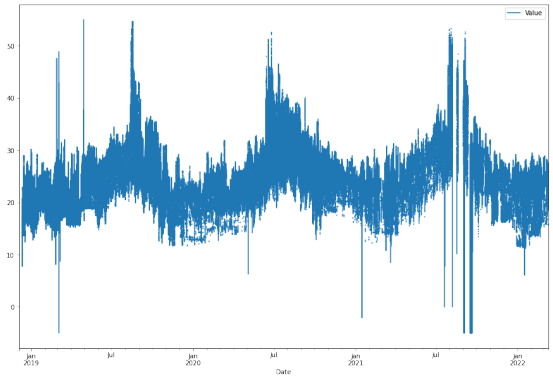
2. Related Works

IoT systems have revolutionized the agricultural production system in recent years. These systems allow a set of sensors to be deployed to monitor different environmental and agronomic variables so that farmers can control their crops. Particularly noteworthy is the technological proposal in greenhouses. Although greenhouses represent a great source of benefits for farmers, as they increase the yield and production of crops, they come with increased responsibility as the climatic conditions of the crops must be controlled and acted upon periodically. Since IoT systems greatly help with this monitoring and acting on the systems, research in this area is increasing [20, 21].

In most applications of IoT systems applied to agriculture, monitoring and control account for most of the research, but prediction for anticipation of problems is still an understudied field [21]. Thus, in [22], the authors propose the use of low-cost irrigation controllers through software embedded in an arduino. The system takes measurements of the air and substrate to ensure optimal plant growth.In [23] a study is carried out on IoT for smart agriculture, analyzing the specific issues and challenges associated with the implementation of IoT to improve agriculture, surveying the wireless communication devices and technologies associated with IoT in agricultural and livestock applications, highlighting the difficulties associated with these solutions. In [24], a microcontroller is developed to control the microchamber of a greenhouse with the aim of increasing the productivity of two varieties of lettuce. The experimental results shows that the designed equipment worked according to the implemented programming algorithm. However, the ventilation, misting and shading actuators did not control the environmental variables, due to their undersizing, although they did control irrigation correctly. In [20], the authors present an IoT system deployed in a greenhouse to enable automated monitoring and control of temperature, humidity and light variables. The IoT system monitors the greenhouse and automatically activates actuators based on current data and pre-determined thresholds. In addition to those mentioned above, there are many applications that can be found which control and monitor greenhouses [25–27]. However, there are very few papers in the literature that design algorithms to forecast climatic variables inside a greenhouse. A study that uses forecasts to anticipate problems in a greenhouse is presented in [28]. The authors detail a real-time decision support system for disease prevention in a greenhouse tomato crop. This system monitors the greenhouse microclimate and then uses a system of rules to identify possible tomato diseases based on the climatic conditions to which the tomatoes have been exposed. Experimental results show that the system increases the effectiveness of climate control, while providing support for the prevention of diseases that are difficult to eradicate. In [29], an investigation to forecast the indoor temperature of a greenhouse is carried out using linear auto regressive models with external input and auto regressive moving average models with external input. Outdoor air temperature and relative humidity, global solar radiation and sky cloudiness are used as input variables for the models. The models are able to describe the greenhouse temperature evolution satisfactorily, but when the greenhouse is being ventilated, they are not accurate due to the nonlinearity of the ventilation strategies. Another work that also uses autoregressive models for temperature prediction is discussed in [30]. The problems and solutions found are similar to those already discussed. In [31], there is another work that considers autoregressive models as being ventilated to predict the indoor temperature of a greenhouse in Thailand. However, in this work, in addition to the regressive models, a neural network is considered to forecast this variable. The results indicate that the neural network in combination with autoregressive models achieves more accurate results. A similar investigation to the previous one but undertaken in Mexico is presented in [32]. In this study, the authors propose a structure of an autoregressive model together with a neural network trained by a Levenberg-Marquardt backpropagation algorithm to forecast the indoor temperature of a greenhouse. To perform this forecasting, the procedure uses external climatic variables. Another work forecasting the indoor temperature of a greenhouse is presented in [33]. To forecast the temperature it uses humidity, indoor temperature and light intensity as input parameters. As part of the research, a BP neural network enhanced with a K-Nearest Neighbor algorithm is applied.



(a) Boxplot of raw data for indoor temperature data from the greenhouse, where you can see a large number of existing outliers (the black circles at the top and bottom of the box plot), the maximum and minimum values (not considered outliers) of the time series (the whiskers of the box plot), the concentration of most of the values (the box plot -quartile 1 and 3-) and the median (the inner line of the box plot -quartile 2-).



(b) Time series of raw data for indoor temperature data from the greenhouse, in which the distribution of values and their trend over time can be observed. The X-axis shows the dates on which the values were collected and the Y-axis shows the degrees Celsius for that particular date.

Fig. 1. Description of the raw greenhouse indoor temperature data collected by the sensors.

As can be seen, there are many studies on control and monitoring, but few very specific ones in terms of forecasting the temperature inside greenhouses. All the studies have a consensus on the importance of this variable. In the works on indoor temperature forecasting, more than one variable is used to carry out such a forecast and not all of them consider temperature data as a time series. Therefore, the design for this work uses only the temperature variable, simplifying the model as we only control and collect data on one variable, allowing the possibility of applying this system in greenhouses where technological resources are usually scarce.

3. Materials and Methods

As previously mentioned, this paper aims to design and optimize four ML techniques to forecast the indoor temperature for an operational greenhouse. This section first describes the dataset and procedures that have been used for data curation. It is important to note that we are interested in demonstrating which ML model is more tolerant to the more than likely presence of noise in the data generated by the greenhouse, so different datasets are generated. It then briefly introduces the four ML techniques used, providing details about how they have been designed and adapted to work with this time series data.

3.1. Data curation and testing dataset preparation

The greenhouse being studied has been in continuous operation since June 2018 and has generated a dataset with more than three years of real operational data for the greenhouse, including up to 21 different variables as previously mentioned. The greenhouse is operating in a semi-arid region under extreme climate conditions, particularly in summer, where temperatures of more than 45 ºC can be reached. It is also important to note that the sensor systems are energy efficient and they therefore reduce the amount of data submissions through the network by not sending

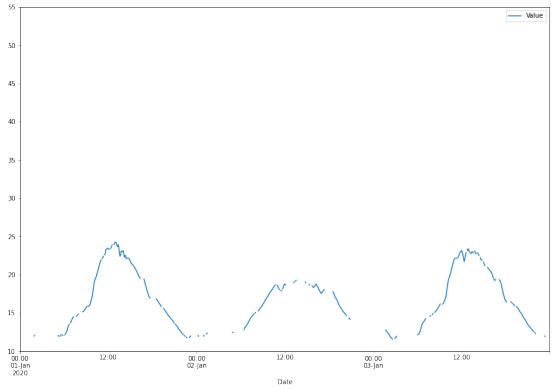
data when there is no change in a particular variable. In other words, if the temperature does not change for a following 5-minute interval, the data is not published to the “Temperature" topic of the MQTT broker. Under these conditions and over such a long sampling period, a large number of errors and null values, among others, are expected to be in the dataset obtained, which may compromise the training procedures of ML models. Indeed, Figure 1 shows a data description of the raw historical data of the greenhouse indoor temperature (TBS). Figure 1a shows the maximum value registered is 55ºC, the minimum value -5ºC, and the average value is 22.9ºC with a clear concentration between 19ºC and 28ºC (see Figure 1b). It is highly unlikely temperatures below 0ºC or above 50ºC can occur in the area where the greenhouse is currently operating, meaning there are few out-of-range values that need to be fixed before carrying out the subsequent steps. As such, several types of action need to be taken for data sanitization to ensure a good starting point for training the ML models.

Figure 2 shows a three-day period to draw attention to the effects of applying all sanitization techniques. Figure 2a shows the raw information as retrived from the greenhouse. It is worth highlighting that there are missing values, so the first step in our sanitization procedure is to fill in the existing gaps due to the low-power mode of the sensor system. This is straightforward as it is deactivated when the temperature inside the greenhouse does not change. Consequently, the backfilling merely consists of replicating the last value collected before the gaps appeared in the time series. This procedure generates the DIRTY dataset (see Figure 2b). The next step is to dig deeper into the raw data generated in the greenhouse by performing an outlier removal and interpolating those values that have been denoted as outliers. The identification of outliers is carried out as follows. First of all, those values that could be considered as outliers need to be identified by applying the Standard Deviation (STD) formula with a threshold of 95%. Additionally, the atypical values need to be identified , i.e. those values that cannot be considered outliers by definition, but are erroneous values within the time-series and, for this reason, a window method is used that compares the current value with the average of the last 3 values and checks that the difference does not exceed 3 degrees in absolute value. Once all the outliers or atypical values have been identified, a data imputation is carried out by performing a linear interpolation of data, i.e. the imputed value is equal to the mean of the previous value and the next value. This step ends with the generation of the CLEAN dataset (see Figure 2c). The last dataset is obtained by smoothing the CLEAN dataset. Several ML techniques benefit from data smoothing [34]. This statistical approach attempts to remove outliers from the data set, making patterns more discernible. During data compilation, data can be altered to reduce or eliminate any wide variations or other statistical noise. Smoothing helps ML models find trends or patterns that would otherwise have been missed. This approach uses simplified enhancements to better forecast various patterns. It focuses on creating a basic direction for the main data points, avoiding any volatile data and drawing a smoother curve through the data points. There are several techniques for data smoothing [35]. Among them, it is worth highlighting Kakman Filters (KF) [36] as it has been widely used for time series smoothing in many domains [37–39]. KF provides a sequential, unbiased, and minimum error variance estimate that works well for discrete-time filtering problems where the underlying physical phenomenon is modeled as a discrete-time process [40]. As a result, a new smooth greenhouse temperature time series was generated as shown in Figure 2d.

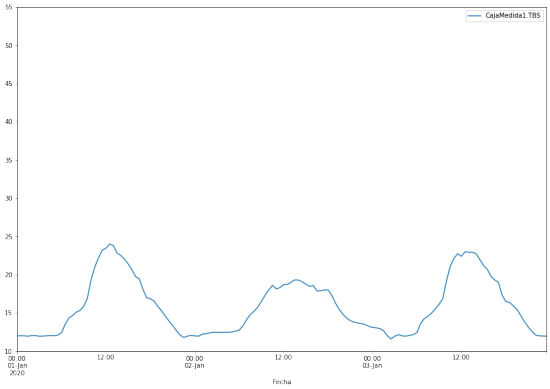
Finally, three temporal granularities are studied. Firstly, the values (i.e., 5-minute temperature samples) included in all the datasets described above are grouped into 15, 30 and 60-minute datasets. This temporal grouping is carried out by applying the arithmetic mean of the values in that interval. Secondly, the datasets are divided into summer (i.e. XXX-SUMMER) and winter (i.e. XXX-WINTER) periods. In semi-arid regions such as southern Spain, the winter periods are much more stable than the summer periods, where the thermal amplitude is much greater. It is important to note that this division is made only for the testing phase of the algorithms. The training is performed on the complete dataset. Table 1 summarizes the description of the datasets that have been used to carry out the temperature forecasting. It shows the start date of the data, the end date and the number of instances contained in each dataset. Each dataset contains the temperature values of a greenhouse between the indicated dates, distinguishing between winter and summer.

3.2. Data transformation

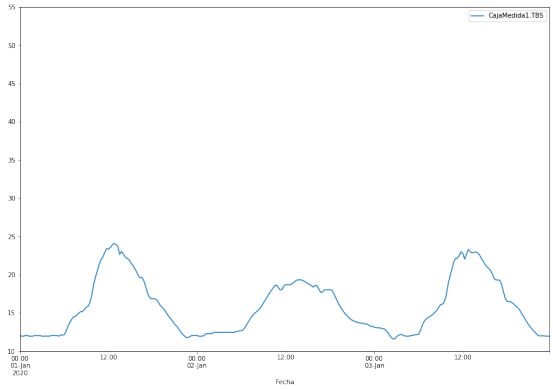
In order to adapt the regression models to time series models, some transformations need to be performed in the dataset, to adapt the time series to a regression problem (supervised problem) with which the model can work. This transformation consists of grouping the data in windows of a certain number of values, for which the output would



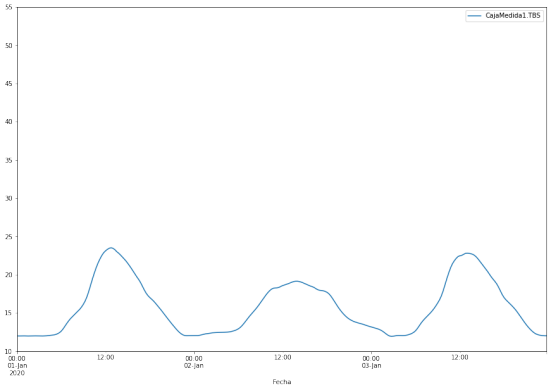
(a) Three day time series for raw data as generated by the greenhouse. At this point, no data pre-processing has been performed. The data are displayed as they are sent and collected from the greenhouse itself.



(b) Three day time series for DIRTY dataset; i.e. after filling the gaps generated by low-power mode. At this point there are no longer any empty values in the time series since they have been filled in by interpolations of data, in order to maintain the characteristics of the original time series.



(c) Three day time series for CLEAN dataset; i.e. after removal of outliers. At this point the data has been cleaned, removing outliers, atypical values, out of range values, filling missing values, etc. Accordingly, the dataset is clean, and optimized for artificial intelligence models.



(d) Three day time series for SMOOTH dataset; i.e. after smoothing. At this point the data have been smoothed by Kalman filters to eliminate possible peaks (noise) that the time series shows, meaning some imperfections that might be found in the data collection are eliminated, although it could also be the case of removing some important information from the data.

Fig. 2. Snapshot of the indoor temperature of the greenhouse, where you can see the changes in and effects of the pre-processing of the data. Since they are collected raw, the empty values are filled in (DIRTY), while the data are cleaned (CLEAN) and smoothed (SMOOTH). A window of 3 random days was chosen to show the effect of data pre-processing.

be the next value to the current window. An example of this approach using windows of three values can be seen in figure 3. Once the data have been transformed into windows, in order to convert the time-series problem into a supervised problem, they can be used as input to the models, “X” being the set of features and “y” the target to be achieved.

3.3. Machine Learning models

This section describes the four machine learning models used:

Datasets	Start date	End Date	# Instances
CLEAN-DS-15-SUMMER	18-12-18	06-06-21	86544
CLEAN-DS-15-WINTER	18-12-18	17-01-21	73104
CLEAN-DS-30-SUMMER	18-12-18	06-06-21	43273
CLEAN-DS-30-WINTER	18-12-18	17-01-21	36553
CLEAN-DS-60-SUMMER	18-12-18	06-06-21	21637
CLEAN-DS-60-WINTER	18-12-18	17-01-21	18277
DIRTY-DS-15-SUMMER	18-12-18	06-06-21	86544
DIRTY-DS-15-WINTER	18-12-18	17-01-21	73104
DIRTY-DS-30-SUMMER	18-12-18	06-06-21	43273
DIRTY-DS-30-WINTER	18-12-18	17-01-21	36553
DIRTY-DS-60-SUMMER	18-12-18	06-06-21	21637
DIRTY-DS-60-WINTER	18-12-18	17-01-21	18277
SMOOTH-DS-15-SUMMER	18-12-18	06-06-21	86544
SMOOTH-DS-15-WINTER	18-12-18	17-01-21	73104
SMOOTH-DS-30-SUMMER	18-12-18	06-06-21	43273
SMOOTH-DS-30-WINTER	18-12-18	17-01-21	36553
SMOOTH-DS-60-SUMMER	18-12-18	06-06-21	21637
SMOOTH-DS-60-WINTER	18-12-18	17-01-21	18277

Table 1
Dataset description

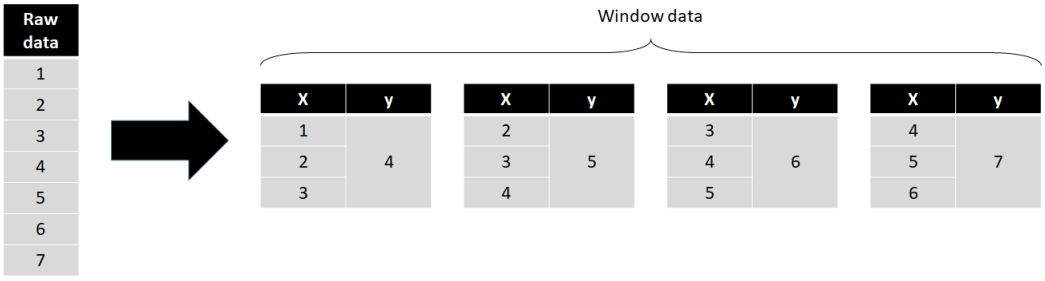


Fig. 3. Data transformation process example using a three values window.

– **Autoregresive (AR)**:This ML model performs prediction by linear combination in a univariate model. An autoregressive model regresses on the same variable being studied, allowing it to handle a wide range of different time series patterns. [41].
– **Autoregressive Integrated Moving Average (ARIMA)**: This Autoregressive model is a linear statistical model that allows regressions of statistical data to find patterns for future prediction ([42]). It is a combination of autoregression (AR) which refers to the lags of the series, while moving average (MA) refers to the lags of the errors and finally integration (I) is the number of differences used to make the time series stationary.
– **Random Forest Regressor (RFR)**: Random Forest Regression (RFR) is an extension of regression decision trees (DTR), which are easy-to-interpret ML models as they generate a model tree that can be easily transformed into decision rules. This method may not be sufficient for the model to learn the characteristics of the model and in addition, DTRs have the problem that they often suffer from overfitting, so multiple decision trees created randomly are often used along with a decision system that allows the model to be improved. It

is precisely this set of trees that forms the Random Forest algorithm as it can be seen as a forest of random trees. This algorithm shows very good results in regression, and together with ANNs, it is widely used for its robustness and speed. [43].

– **K-Nearest Neighbors Regressor (KNNR)**: This algorithm is based on the similarity of sample characteristics, such that a new sample is assigned a value based on its similarity to the samples in the training set. Initially, the distance between the new point and each training point is calculated. For this work, the distance has been calculated using the Euclidean distance formula, but it can be calculated by other methods, such as the Manhattan distance formula. This algorithm is rather more popular for classification problems, although it is also used in regression problems, called KNNR. A disadvantage of these methods is that the number of neighbours (k) to be considered with respect to the new sample has to be set. In our case, K=24 has been used. The choice of the value of K is important, because if it is a very small value there may be overfitting, i.e. the classification is too close to the training set. Conversely, a very high value will make a poorly trained model. [44].

4. Evaluation and Discussion

In this section, an experiment is carried out, namely, the evaluation of the 5 machine learning techniques proposed in section 3.3. For each technique, the noise tolerance is evaluated and analyzed individually, as well as the 12 and 24 hour forecasting results, taking into account the temporal granularity. Furthermore, a comparison of results using the four techniques is performed to analyze the most accurate technique for indoor temperature forecasting. The measures used for the evaluation in each of the experiments are Root Mean Square Error (`RMSE`), Mean Absolute Error (`MAE`) and coefficient of determination (R^2).

4.1. Metrics

In assessing the quality, robustness and accuracy of the models, the following metrics are used, where y_i is the real data for instance i, p_i is the forecast for instance i and N is the total number of forecasted instances. :

– **Coefficient of determination ($\mathbf{R}^2$)**:This coefficient focuses on analyzing the differences between the output variable and the predictor variable. Its possible range of values is between 0 and 1, with the best models being those with a coefficient closer to 1 [45].

$$R^2 = \frac{(\sum_{i=1}^{N}(y_i-\bar{y})(p_i-\bar{p}))^2}{\sum_{i=1}^{N}(y_i-\bar{y})^2\sum_{i=1}^{N}(p_i-\bar{p})^2} \tag{1}$$

– **Root mean square error (RMSE)**: This value measures the amount of error between two sets of data. In other words, it compares a predicted value and an observed or known value. It is calculated as the square root of a variance. RMSE can be interpreted as the standard deviation of the unexplained variance, and has the useful property of being in the same units as the response variable. Lower values of RMSE indicate a better fit. [46].

$$RMSE = \sqrt{\frac{\sum_{i=1}^{N}(y_i-p_i)^2}{N}} \tag{2}$$

– **Mean absolute error (MAE)**: This value is a measure of the difference between two continuous variables. Considering two sets of data (some calculated and some observed) related to the same phenomenon, the mean absolute error is used to quantify the accuracy of a prediction technique. [46].

$$MAE = \frac{\sum_{i=1}^{N}|y_i-p_i|}{N} \tag{3}$$

4.2. Machine learning model evaluation

This section shows the results of the different ML techniques for all datasets previously described in Table 1. These results are broadly discussed in Section 4.

Forecasting period	12h			24h		
Datasets	R^2_{sd}	$RMSE_{sd}$	MAE_{sd}	R^2_{sd}	$RMSE_{sd}$	MAE_{sd}
CLEAN-DS-15-SUMMER	$0.196_{0.000}$	$3.691_{0.000}$	$2.625_{0.000}$	$0.675_{0.000}$	$5.549_{0.000}$	$4.446_{0.000}$
CLEAN-DS-15-WINTER	$0.121_{0.000}$	$3.024_{0.000}$	$2.697_{0.000}$	$0.334_{0.000}$	$3.316_{0.000}$	$2.923_{0.000}$
CLEAN-DS-30-SUMMER	$0.022_{0.000}$	$3.144_{0.000}$	$2.366_{0.000}$	$0.794_{0.000}$	$4.955_{0.000}$	$4.108_{0.000}$
CLEAN-DS-30-WINTER	$0.003_{0.000}$	$3.434_{0.000}$	$3.018_{0.000}$	$0.298_{0.000}$	$3.329_{0.000}$	$2.940_{0.000}$
CLEAN-DS-60-SUMMER	$0.161_{0.000}$	$3.235_{0.000}$	$2.709_{0.000}$	$0.859_{0.000}$	$3.579_{0.000}$	$3.164_{0.000}$
CLEAN-DS-60-WINTER	$0.021_{0.000}$	$1.701_{0.000}$	$1.524_{0.000}$	$0.914_{0.000}$	$1.657_{0.000}$	$1.334_{0.000}$
DIRTY-DS-15-SUMMER	$0.193_{0.000}$	$3.625_{0.000}$	$2.621_{0.000}$	$0.674_{0.000}$	$5.392_{0.000}$	$4.323_{0.000}$
DIRTY-DS-15-WINTER	$0.121_{0.000}$	$3.110_{0.000}$	$2.773_{0.000}$	$0.335_{0.000}$	$3.342_{0.000}$	$2.940_{0.000}$
DIRTY-DS-30-SUMMER	$0.021_{0.000}$	$3.156_{0.000}$	$2.420_{0.000}$	$0.767_{0.000}$	$4.915_{0.000}$	$4.083_{0.000}$
DIRTY-DS-30-WINTER	$0.000_{0.000}$	$3.506_{0.000}$	$3.076_{0.000}$	$0.278_{0.000}$	$3.381_{0.000}$	$2.999_{0.000}$
DIRTY-DS-60-SUMMER	$0.160_{0.000}$	$3.240_{0.000}$	$2.726_{0.000}$	$0.859_{0.000}$	$3.582_{0.000}$	$3.171_{0.000}$
DIRTY-DS-60-WINTER	$0.023_{0.000}$	$1.723_{0.000}$	$1.529_{0.000}$	$0.911_{0.000}$	$1.665_{0.000}$	$1.335_{0.000}$
SMOOTH-DS-15-SUMMER	$0.149_{0.000}$	$3.615_{0.000}$	$2.603_{0.000}$	$0.733_{0.000}$	$5.518_{0.000}$	$4.448_{0.000}$
SMOOTH-DS-15-WINTER	$0.117_{0.000}$	$2.835_{0.000}$	$2.463_{0.000}$	$0.325_{0.000}$	$3.267_{0.000}$	$2.843_{0.000}$
SMOOTH-DS-30-SUMMER	$0.058_{0.000}$	$2.851_{0.000}$	$2.129_{0.000}$	$0.924_{0.000}$	$4.218_{0.000}$	$3.544_{0.000}$
SMOOTH-DS-30-WINTER	$0.000_{0.000}$	$2.737_{0.000}$	$2.313_{0.000}$	$0.624_{0.000}$	$2.647_{0.000}$	$2.225_{0.000}$
SMOOTH-DS-60-SUMMER	$0.164_{0.000}$	$3.240_{0.000}$	$2.513_{0.000}$	$0.900_{0.000}$	$4.183_{0.000}$	$3.562_{0.000}$
SMOOTH-DS-60-WINTER	$0.026_{0.000}$	$1.981_{0.000}$	$1.735_{0.000}$	$0.874_{0.000}$	$2.117_{0.000}$	$1.709_{0.000}$

Table 2

Results of the AR technique, values in the sub-index indicate the standard deviation obtained after the repetition of each experiment. R^2 (coefficient of determination), RMSE (root mean square error) and MAE (mean absolute error) are the metrics used in the results acquisition. RMSE and MAE are measured in degrees Celsius (°C).

Table 2 shows the results obtained for all datasets using the AR model. Before going any further, it is important to highlight that the AR model is a technique that has no random components, and thus the standard deviation must be zero. The AR model obtains the worst results in terms of R^2 for all targeted ML techniques. However, the RMSE and MAE values maintain their values at a reasonable threshold; i.e. around 3°C error. The best result is obtained with the CLEAN-DS-60-WINTER and DIRTY-DS-60-WINTER datasets, reaching RMSE and MAE figures of around 1.7°C and 1.5°C respectively. It is also important to note that the 24-hour forecast obtains better metrics with respect to R^2 and maintains the values in MAE and RMSE.

Table 3 shows the results obtained for all datasets using the ARIMA model. Before going any further, it should be remembered that since it is a technique that has no random components, the standard deviation is zero like in the AR case. On the other hand, the ARIMA model obtains results that are not very stable for certain datasets. In general, it obtains more accurate results for the 24-hour forecast than for the 12-hour forecast and has a lower RMSE and MAE for CLEAN-DS-60-WINTER and DIRTY-DS-60-WINTER datasets, the error being around 1.4°C and 1.6°C.

Table 4 shows the results obtained for all datasets using the KNNR model. Before going any further, it should be noted that this is a technique using an instance-based learning model and has no random component, so the standard deviation is again zero. Regarding the results, KNNR obtains stable models, with few differences between 12-hour and 24-hour forecasting accuracy metrics. In addition, the best result in both MAE and RMSE is shown by forecasting 12-hours using the CLEAN-DS-15-WINTER dataset, obtaining RMSE and MAE values of 1.241°C and 1.021°C respectively.

Table 5 shows results obtained for all datasets using the RFR model. This technique obtains very satisfactory results with an RMSE and MAE below 0.5°C for all datasets; i.e. DIRTY, CLEAN and SMOOTH datasets. The

Forecasting period	12h			24h		
Datasets	R^2_{sd}	$RMSE_{sd}$	MAE_{sd}	R^2_{sd}	$RMSE_{sd}$	MAE_{sd}
CLEAN-DS-15-SUMMER	$0.245_{0.000}$	$11.548_{0.000}$	$8.055_{0.000}$	$0.307_{0.000}$	$27.547_{0.000}$	$22.435_{0.000}$
CLEAN-DS-15-WINTER	$0.270_{0.000}$	$9.080_{0.000}$	$7.670_{0.000}$	$0.036_{0.000}$	$38.007_{0.000}$	$27.882_{0.000}$
CLEAN-DS-30-SUMMER	$0.243_{0.000}$	$12.240_{0.000}$	$8.728_{0.000}$	$0.319_{0.000}$	$28.496_{0.000}$	$23.374_{0.000}$
CLEAN-DS-30-WINTER	$0.246_{0.000}$	$4.491_{0.000}$	$2.564_{0.000}$	$0.104_{0.000}$	$7.739_{0.000}$	$6.101_{0.000}$
CLEAN-DS-60-SUMMER	$0.103_{0.000}$	$4.430_{0.000}$	$2.623_{0.000}$	$0.559_{0.000}$	$6.068_{0.000}$	$4.939_{0.000}$
CLEAN-DS-60-WINTER	$0.011_{0.000}$	$1.470_{0.000}$	$1.322_{0.000}$	$0.939_{0.000}$	$1.429_{0.000}$	$1.169_{0.000}$
DIRTY-DS-15-SUMMER	$0.244_{0.000}$	$11.142_{0.000}$	$7.772_{0.000}$	$0.315_{0.000}$	$25.840_{0.000}$	$21.140_{0.000}$
DIRTY-DS-15-WINTER	$0.270_{0.000}$	$22.258_{0.000}$	$18.159_{0.000}$	$0.038_{0.000}$	$83.834_{0.000}$	$63.202_{0.000}$
DIRTY-DS-30-SUMMER	$0.033_{0.000}$	$4.434_{0.000}$	$3.435_{0.000}$	$0.226_{0.000}$	$6.696_{0.000}$	$5.709_{0.000}$
DIRTY-DS-30-WINTER	$0.254_{0.000}$	$5.663_{0.000}$	$3.502_{0.000}$	$0.093_{0.000}$	$10.770_{0.000}$	$8.781_{0.000}$
DIRTY-DS-60-SUMMER	$0.122_{0.000}$	$4.294_{0.000}$	$2.645_{0.000}$	$0.596_{0.000}$	$5.803_{0.000}$	$4.763_{0.000}$
DIRTY-DS-60-WINTER	$0.003_{0.000}$	$1.922_{0.000}$	$1.673_{0.000}$	$0.880_{0.000}$	$1.674_{0.000}$	$1.341_{0.000}$
SMOOTH-DS-15-SUMMER	$0.249_{0.000}$	$2.326_{0.000}$	$1.691_{0.000}$	$0.191_{0.000}$	$18.595_{0.000}$	$11.473_{0.000}$
SMOOTH-DS-15-WINTER	$0.269_{0.000}$	$4.126_{0.000}$	$2.416_{0.000}$	$0.029_{0.000}$	$8.911_{0.000}$	$7.132_{0.000}$
SMOOTH-DS-30-SUMMER	$0.238_{0.000}$	$9.814_{0.000}$	$6.527_{0.000}$	$0.347_{0.000}$	$21.423_{0.000}$	$17.535_{0.000}$
SMOOTH-DS-30-WINTER	$0.215_{0.000}$	$3.966_{0.000}$	$2.219_{0.000}$	$0.233_{0.000}$	$6.396_{0.000}$	$4.824_{0.000}$
SMOOTH-DS-60-SUMMER	$0.032_{0.000}$	$4.111_{0.000}$	$2.679_{0.000}$	$0.490_{0.000}$	$6.247_{0.000}$	$5.181_{0.000}$
SMOOTH-DS-60-WINTER	$0.040_{0.000}$	$1.828_{0.000}$	$1.580_{0.000}$	$0.901_{0.000}$	$2.055_{0.000}$	$1.640_{0.000}$

Table 3

Results of the ARIMA technique, values in the sub-index indicate the standard deviation obtained after the repetition of each experiment. R^2 (coefficient of determination), RMSE (root mean square error) and MAE (mean absolute error) are the metrics used in the results acquisition. RMSE and MAE are measured in degrees Celsius (°C).

worst-performing dataset for any preprocessing is the summer dataset with a time granularity of 60 minutes. Despite delivering worse results, we still have an RMSE and MAE of around 1°C.

The analyses of the results were statistically validated using Friedman's test [47] for two-to-two comparisons with the results of Friedman's test before being adjusted with Dun Bonferroni's post hoc test [48]. In the statistical tests, the MAE value was used. The first statistical analysis was carried out to find out whether there are significant differences between the 12-hour and 24-hour forecasts. This analysis indicates with a confidence level of 95% that there are no significant differences between the techniques when forecasting at 12 and 24 hours.

4.3. Discussion

Analysing whether there are significant differences between the results when considering the type of preprocessing, the Bonferroni test indicates that with a 95% confidence level, there are significant differences between results achieved by ML methods with DIRTY and SMOOTH datasets and between CLEAN and SMOOTH datasets, with p-values of 0.001 and 0.0 respectively. However, there are no significant differences between the results when using DIRTY and CLEAN datasets. This suggests that our DIRTY dataset is good enough and does not need any additional preprocessing to obtain good forecasting accuracy.

Table 6 shows the results of the p-values and the p-values adjusted with the Bonferroni test to identify whether there are significant differences between the different techniques. Only p-values with significant differences are shown. The statistical tests indicate with a 95% confidence level that, for the MAE value, the best performance is for the KNNR and RFR techniques, as there are no significant differences between them. The RFR technique obtains better results, i.e. lower MAE, than the other techniques (ARIMA and AR).

Results achieved by all targeted ML techniques are satisfactory and their performance is stable and robust. However, there are some techniques that perform better in the general scenario. In particular, the worst performing techniques are the autoregressive methods; i.e. the AR and ARIMA techniques falling a long way short of the KNN and RFR, which perform much better in this case. In general terms, the MAE and RMSE error metrics are around 1-1.5°C, except for RFR where the result is a little lower, obtaining a small standard deviation and error values

Forecasting period	12h			24h		
Datasets	R^2_{sd}	$RMSE_{sd}$	MAE_{sd}	R^2_{sd}	$RMSE_{sd}$	MAE_{sd}
CLEAN-DS-15-SUMMER	$0.771_{0.000}$	$3.485_{0.000}$	$2.736_{0.000}$	$0.788_{0.000}$	$3.938_{0.000}$	$3.421_{0.000}$
CLEAN-DS-15-WINTER	$0.979_{0.000}$	$1.241_{0.000}$	$1.021_{0.000}$	$0.973_{0.000}$	$1.470_{0.000}$	$1.151_{0.000}$
CLEAN-DS-30-SUMMER	$0.911_{0.000}$	$3.258_{0.000}$	$2.472_{0.000}$	$0.850_{0.000}$	$3.813_{0.000}$	$3.285_{0.000}$
CLEAN-DS-30-WINTER	$0.992_{0.000}$	$1.403_{0.000}$	$1.133_{0.000}$	$0.966_{0.000}$	$1.639_{0.000}$	$1.247_{0.000}$
CLEAN-DS-60-SUMMER	$0.887_{0.000}$	$3.211_{0.000}$	$2.470_{0.000}$	$0.860_{0.000}$	$3.748_{0.000}$	$3.234_{0.000}$
CLEAN-DS-60-WINTER	$0.964_{0.000}$	$1.347_{0.000}$	$1.019_{0.000}$	$0.954_{0.000}$	$1.662_{0.000}$	$1.201_{0.000}$
DIRTY-DS-15-SUMMER	$0.797_{0.000}$	$3.474_{0.000}$	$2.689_{0.000}$	$0.793_{0.000}$	$3.927_{0.000}$	$3.394_{0.000}$
DIRTY-DS-15-WINTER	$0.972_{0.000}$	$1.244_{0.000}$	$0.980_{0.000}$	$0.970_{0.000}$	$1.488_{0.000}$	$1.147_{0.000}$
DIRTY-DS-30-SUMMER	$0.912_{0.000}$	$3.259_{0.000}$	$2.457_{0.000}$	$0.850_{0.000}$	$3.808_{0.000}$	$3.273_{0.000}$
DIRTY-DS-30-WINTER	$0.991_{0.000}$	$1.366_{0.000}$	$1.104_{0.000}$	$0.968_{0.000}$	$1.648_{0.000}$	$1.249_{0.000}$
DIRTY-DS-60-SUMMER	$0.919_{0.000}$	$3.219_{0.000}$	$2.466_{0.000}$	$0.858_{0.000}$	$3.792_{0.000}$	$3.268_{0.000}$
DIRTY-DS-60-WINTER	$0.958_{0.000}$	$1.345_{0.000}$	$1.024_{0.000}$	$0.947_{0.000}$	$1.693_{0.000}$	$1.221_{0.000}$
SMOOTH-DS-15-SUMMER	$0.860_{0.000}$	$3.329_{0.000}$	$2.657_{0.000}$	$0.833_{0.000}$	$3.886_{0.000}$	$3.396_{0.000}$
SMOOTH-DS-15-WINTER	$0.961_{0.000}$	$1.397_{0.000}$	$1.105_{0.000}$	$0.963_{0.000}$	$1.591_{0.000}$	$1.214_{0.000}$
SMOOTH-DS-30-SUMMER	$0.935_{0.000}$	$3.108_{0.000}$	$2.493_{0.000}$	$0.879_{0.000}$	$3.734_{0.000}$	$3.284_{0.000}$
SMOOTH-DS-30-WINTER	$0.930_{0.000}$	$1.508_{0.000}$	$1.165_{0.000}$	$0.946_{0.000}$	$1.817_{0.000}$	$1.358_{0.000}$
SMOOTH-DS-60-SUMMER	$0.934_{0.000}$	$3.065_{0.000}$	$2.540_{0.000}$	$0.917_{0.000}$	$3.862_{0.000}$	$3.410_{0.000}$
SMOOTH-DS-60-WINTER	$0.839_{0.000}$	$1.959_{0.000}$	$1.755_{0.000}$	$0.905_{0.000}$	$2.267_{0.000}$	$1.821_{0.000}$

Table 4

Results of the KNNR technique, values in the sub-index indicate the standard deviation obtained after the repetition of each experiment. R^2 (coefficient of determination), RMSE (root mean square error) and MAE (mean absolute error) are the metrics used in the results acquisition. RMSE and MAE are measured in degrees Celsius (°C).

below 1°C. It is also important to note that most techniques do not show large differences between the 12-hour and 24-hour forecasting.

Regarding the dataset curation procedure impact on performance of ML methods, results obtained by using DIRTY and CLEAN datasets are very similar. The CLEAN dataset may offer slightly better performance in ML techniques, while The SMOOTH dataset always offers worse performance than its counterpart version (i.e., CLEAN and DIRTY) because the Kalman filter used removes relevant information that is necessary for these methods to learn and hence the results are more unstable and with a greater forecasting error.

It is also important to note that datasets evaluated in winter always perform better than those evaluated during the summer. This is due to the exponential variation and increase in summer temperatures within a short time. When temperatures are moderate, ML techniques perform very well, but when they are very high, the techniques result in a greater error. This is not a relevant problem, since in greenhouses the indoor temperatures to be forecasted are moderate and in these cases the techniques perform very satisfactorily.

Summing up, the RFR technique can be considered the best performer for forecasting temperature at both 12 and 24 hours. In addition, the SMOOTH curation procedure performs worse in all ML techniques than the other two curation procedures (DIRTY and CLEAN), whose results do not differ significantly. Taking this into account, we conclude that it is not necessary to perform preprocessing of the data to obtain good forecasting accuracy, this is to obtain low prediction error.

Finally, it is important to note that this work uses a methodology that can be applied independently of the technology or techniques used. In this methodology, the main steps consist of obtaining a data collection, initially to create a history of the data and subsequently to carry out analysis and inference. After having the information collection system, it is important to have a pre-processing method to ensure the quality of the data. Subsequently, analysis techniques and pattern extraction from the data are interesting to obtain information for decision making by ML models. Lastly, once the best prediction model and the best granularity of both temporal and long-term prediction have been analysed, such a model could then be implemented in a decision support system.

Forecasting period	12h			24h		
Datasets	R^2_{sd}	$RMSE_{sd}$	MAE_{sd}	R^2_{sd}	$RMSE_{sd}$	MAE_{sd}
CLEAN-DS-15-SUMMER	$0.900_{0.050}$	$3.183_{0.181}$	$2.116_{0.132}$	$0.872_{0.035}$	$3.672_{0.115}$	$2.961_{0.088}$
CLEAN-DS-15-WINTER	$0.929_{0.016}$	$1.636_{0.109}$	$1.414_{0.057}$	$0.929_{0.021}$	$1.708_{0.187}$	$1.339_{0.123}$
CLEAN-DS-30-SUMMER	$0.964_{0.007}$	$2.628_{0.062}$	$1.855_{0.050}$	$0.939_{0.008}$	$3.195_{0.019}$	$2.671_{0.018}$
CLEAN-DS-30-WINTER	$0.931_{0.003}$	$1.431_{0.019}$	$1.177_{0.023}$	$0.954_{0.001}$	$1.626_{0.006}$	$1.243_{0.009}$
CLEAN-DS-60-SUMMER	$0.971_{0.012}$	$2.377_{0.247}$	$1.701_{0.115}$	$0.942_{0.016}$	$2.820_{0.135}$	$2.366_{0.084}$
CLEAN-DS-60-WINTER	$0.966_{0.003}$	$1.111_{0.051}$	$0.788_{0.024}$	$0.967_{0.003}$	$1.373_{0.039}$	$0.956_{0.021}$
DIRTY-DS-15-SUMMER	$0.860_{0.081}$	$3.273_{0.215}$	$2.173_{0.171}$	$0.857_{0.044}$	$3.719_{0.101}$	$2.998_{0.081}$
DIRTY-DS-15-WINTER	$0.954_{0.011}$	$1.661_{0.079}$	$1.399_{0.038}$	$0.928_{0.016}$	$1.772_{0.158}$	$1.368_{0.099}$
DIRTY-DS-30-SUMMER	$0.931_{0.016}$	$2.858_{0.193}$	$1.981_{0.145}$	$0.913_{0.020}$	$3.332_{0.126}$	$2.758_{0.092}$
DIRTY-DS-30-WINTER	$0.898_{0.019}$	$1.546_{0.072}$	$1.299_{0.075}$	$0.942_{0.005}$	$1.704_{0.037}$	$1.332_{0.045}$
DIRTY-DS-60-SUMMER	$0.966_{0.002}$	$2.070_{0.038}$	$1.521_{0.022}$	$0.960_{0.001}$	$2.628_{0.025}$	$2.225_{0.020}$
DIRTY-DS-60-WINTER	$0.972_{0.007}$	$0.915_{0.095}$	$0.705_{0.085}$	$0.974_{0.003}$	$1.131_{0.071}$	$0.829_{0.057}$
SMOOTH-DS-15-SUMMER	$0.899_{0.103}$	$2.437_{0.596}$	$1.711_{0.310}$	$0.924_{0.059}$	$2.976_{0.363}$	$2.454_{0.238}$
SMOOTH-DS-15-WINTER	$0.901_{0.018}$	$1.607_{0.089}$	$1.413_{0.055}$	$0.934_{0.014}$	$1.737_{0.100}$	$1.410_{0.056}$
SMOOTH-DS-30-SUMMER	$0.917_{0.024}$	$2.850_{0.175}$	$1.991_{0.091}$	$0.907_{0.020}$	$3.191_{0.195}$	$2.650_{0.124}$
SMOOTH-DS-30-WINTER	$0.781_{0.037}$	$1.691_{0.126}$	$1.355_{0.103}$	$0.879_{0.020}$	$1.623_{0.127}$	$1.267_{0.093}$
SMOOTH-DS-60-SUMMER	$0.941_{0.004}$	$2.977_{0.048}$	$2.250_{0.020}$	$0.917_{0.006}$	$3.625_{0.065}$	$3.062_{0.042}$
SMOOTH-DS-60-WINTER	$0.893_{0.009}$	$1.593_{0.031}$	$1.168_{0.019}$	$0.910_{0.006}$	$1.935_{0.108}$	$1.399_{0.082}$

Table 5

Results of the RFR technique, values in the sub-index indicate the standard deviation obtained after the repetition of each experiment. R^2 (coefficient of determination), RMSE (root mean square error) and MAE (mean absolute error) are the metrics used in the results acquisition. RMSE and MAE are measured in degrees Celsius (°C).

	RRF-ARIMA	RRF-AR	KNN-AR
p-value	0.000	0.000	0.000
p-value adj.	0.000	0.000	0.001

Table 6

Results of the paired Friedman's statistical test (p-value row) and p-value adjustment with the Bonferroni test to find the best performing techniques.

5. Conclusion and Future Work

Intelligent agriculture in greenhouses is currently a highly researched topic. This is due to the fact that its application encompasses different factors including cost optimization, resource optimization as well as increased sustainability. In this work, we tackled the problem of forecasting the indoor temperature in the greenhouse in advance, armed with the knowledge of the temperatures for previous time slots. This was done by applying machine learning techniques to find the most accurate models, with the lowest error, when predicting the temperature 12 and 24 hours in advance. In addition, we also performed three types of data preprocessing to analyse the best of them, always aiming to improve the quality of the data and indirectly the accuracy of the models. After data preprocessing, a total of 5 machine learning techniques were evaluated to analyse the best indoor temperature forecasting model for the greenhouse. After collecting and processing the results, a statistical analysis was carried out which concludes that preprocessing the data does not improve the results, i.e. that the forecasts at 12 and 24 hours are equally reliable and that the technique which obtains the best results is Random Forest Regressor.

As for future work, we have the possibility of including new variables to reduce the temperature forecasting error by creating multivariate models, as well as making use of the actuators that optimize the greenhouse temperature in these forecasts,.

Declarations

Ethical Approval

Not applicable

Conflict of interest

The authors declare they do not have any conflict of interest.

Authors' contributions

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

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, as well as the Ramon y Cajal Grant RYC2018-025580-I, funded by MCIN/AEI/10.13039/501100011033, "FSE invest in your future" and "ERDF A way of making Europe".

References

[1] R. Gebbers and V.I. Adamchuk, Precision agriculture and food security, *Science* **327**(5967) (2010), 828–831.
[2] Y. Achour, A. Ouammi and D. Zejli, Technological progresses in modern sustainable greenhouses cultivation as the path towards precision agriculture, *Renewable and Sustainable Energy Reviews* **147** (2021), 111251.
[3] M. Ghoulem, K. El Moueddeb, E. Nehdi, R. Boukhanouf and J.K. Calautit, Greenhouse design and cooling technologies for sustainable food cultivation in hot climates: Review of current practice and future status, *Biosystems Engineering* **183** (2019), 121–150.
[4] H. Ritchie and M. Roser, CO_2 and Greenhouse Gas Emissions, *Our World in Data* (2020), https://ourworldindata.org/co2-and-other-greenhouse-gas-emissions.
[5] S. Gorjian, F. Calise, K. Kant, M.S. Ahamed, B. Copertaro, G. Najafi, X. Zhang, M. Aghaei and R.R. Shamshiri, A review on opportunities for implementation of solar energy technologies in agricultural greenhouses, *Journal of Cleaner Production* **285** (2021), 124807.
[6] M. Soussi, M.T. Chaibi, M. Buchholz and Z. Saghrouni, Comprehensive Review on Climate Control and Cooling Systems in Greenhouses under Hot and Arid Conditions, *Agronomy* **12**(3) (2022), 626.
[7] W.-H. Chen and F. You, Smart greenhouse control under harsh climate conditions based on data-driven robust model predictive control with principal component analysis and kernel density estimation, *Journal of Process Control* **107** (2021), 103–113.
[8] M. Shang, G. Tian, L. Qin, J. Zhao, H. Ruan and F. Wang, Greenhouse wireless monitoring system based on the zigbee, in: *International Conference on Computer and Computing Technologies in Agriculture*, Springer, 2013, pp. 109–117.
[9] H. Li, Y. Guo, H. Zhao, Y. Wang and D. Chow, Towards automated greenhouse: A state of the art review on greenhouse monitoring methods and technologies based on internet of things, *Computers and Electronics in Agriculture* **191** (2021), 106558.
[10] R. Rayhana, G. Xiao and Z. Liu, Internet of things empowered smart greenhouse farming, *IEEE Journal of Radio Frequency Identification* **4**(3) (2020), 195–211.
[11] Y. Kaluarachchi, Potential advantages in combining smart and green infrastructure over silo approaches for future cities, *Frontiers of Engineering Management* **8**(1) (2021), 98–108.
[12] A. Escamilla-García, G.M. Soto-Zarazúa, M. Toledano-Ayala, E. Rivas-Araiza and A. Gastélum-Barrios, Applications of artificial neural networks in greenhouse technology and overview for smart agriculture development, *Applied Sciences* **10**(11) (2020), 3835.
[13] B. Alhnaity, S. Pearson, G. Leontidis and S. Kollias, Using deep learning to predict plant growth and yield in greenhouse environments, in: *International Symposium on Advanced Technologies and Management for Innovative Greenhouses: GreenSys2019 1296*, 2019, pp. 425–432.
[14] S. Revathi, T.K. Radhakrishnan and N. Sivakumaran, Climate control in greenhouse using intelligent control algorithms, in: *2017 American Control Conference (ACC)*, IEEE, 2017, pp. 887–892.

[15] M. Taki, S.A. Mehdizadeh, A. Rohani, M. Rahnama and M. Rahmati-Joneidabad, Applied machine learning in greenhouse simulation; new application and analysis, *Information processing in agriculture* **5**(2) (2018), 253–268.
[16] A.F. Subahi and K.E. Bouazza, An intelligent IoT-based system design for controlling and monitoring greenhouse temperature, *IEEE Access* **8** (2020), 125488–125500.
[17] A. Tzounis, N. Katsoulas, T. Bartzanas and C. Kittas, Internet of Things in agriculture, recent advances and future challenges, *Biosystems engineering* **164** (2017), 31–48.
[18] M.A. Guillén-Navarro, R. Martínez-España, B. López and J.M. Cecilia, A high-performance IoT solution to reduce frost damages in stone fruits, *Concurrency and Computation: Practice and Experience* **33**(2) (2021), e5299.
[19] A. Castañeda-Miranda and V.M. Castaño-Meneses, Internet of things for smart farming and frost intelligent control in greenhouses, *Computers and Electronics in Agriculture* **176** (2020), 105614.
[20] J.S. Raj and J.V. Ananthi, Automation using IoT in greenhouse environment, *Journal of Information Technology* **1**(01) (2019), 38–47.
[21] J.M. Talavera, L.E. Tobón, J.A. Gómez, M.A. Culman, J.M. Aranda, D.T. Parra, L.A. Quiroz, A. Hoyos and L.E. Garreta, Review of IoT applications in agro-industrial and environmental fields, *Computers and Electronics in Agriculture* **142** (2017), 283–297.
[22] Y.A. Rivas-Sánchez, M.F. Moreno-Pérez and J. Roldán-Cañas, Environment control with low-cost microcontrollers and microprocessors: Application for green walls, *Sustainability* **11**(3) (2019), 782.
[23] P.P. Ray, Internet of things for smart agriculture: Technologies, practices and future direction, *Journal of Ambient Intelligence and Smart Environments* **9**(4) (2017), 395–420.
[24] A.C. Marques Filho, J.P. Rodrigues, S.D.S. de Medeiros and S.R.R. de Medeiros, Development of an electronic controller for lettuce production in greenhouses, *Revista de Agricultura Neotropical* **7**(3) (2020), 65–72.
[25] M.A. Zamora-Izquierdo, J. Santa, J.A. Martínez, V. Martínez and A.F. Skarmeta, Smart farming IoT platform based on edge and cloud computing, *Biosystems engineering* **177** (2019), 4–17.
[26] A. Pawlowski, J. Sánchez-Molina, J. Guzmán, F. Rodríguez and S. Dormido, Evaluation of event-based irrigation system control scheme for tomato crops in greenhouses, *Agricultural Water Management* **183** (2017), 16–25.
[27] M.A. Akkaş and R. Sokullu, An IoT-based greenhouse monitoring system with Micaz motes, *Procedia computer science* **113** (2017), 603–608.
[28] J. Cañadas, J.A. Sánchez-Molina, F. Rodríguez and I.M. del Águila, Improving automatic climate control with decision support techniques to minimize disease effects in greenhouse tomatoes, *Information Processing in Agriculture* **4**(1) (2017), 50–63.
[29] H.U. Frausto, J. Pieters and J. Deltour, Modelling greenhouse temperature by means of auto regressive models, *Biosystems Engineering* **84**(2) (2003), 147–157.
[30] W. Zhang, R. Zhou, C. Zhu et al., Greenhouse temperature simulation based on ARX model., *Jiangsu Journal of Agricultural Sciences* **29**(1) (2013), 46–50.
[31] S. Patil, H. Tantau and V. Salokhe, Modelling of tropical greenhouse temperature by auto regressive and neural network models, *Biosystems engineering* **99**(3) (2008), 423–431.
[32] A. Castañeda-Miranda and V.M. Castaño, Smart frost control in greenhouses by neural networks models, *Computers and Electronics in Agriculture* **137** (2017), 102–114.
[33] X. Li, X. Zhang, Y. Wang, Y.-f. Chen et al., Temperature prediction model for solar greenhouse based on improved BP neural network, in: *Journal of Physics: Conference Series*, Vol. 1639, IOP Publishing, 2020, p. 012036.
[34] F. Terroso-Sáenz, A. Muñoz, J. Fernández-Pedauye and J.M. Cecilia, Human Mobility Prediction With Region-Based Flows and Water Consumption, *IEEE Access* **9** (2021), 88651–88663.
[35] S.F. Chen and R. Rosenfeld, A survey of smoothing techniques for ME models, *IEEE transactions on Speech and Audio Processing* **8**(1) (2000), 37–50.
[36] Y. Kim and H. Bang, Introduction to Kalman filter and its applications, *Introduction and Implementations of the Kalman Filter* **1** (2018), 1–16.
[37] L. Ralaivola and F. d'Alché-Buc, Time series filtering, smoothing and learning using the kernel Kalman filter, in: *Proceedings. 2005 IEEE International Joint Conference on Neural Networks, 2005.*, Vol. 3, IEEE, 2005, pp. 1449–1454.
[38] F. Avanzi, Z. Zheng, A. Coogan, R. Rice, R. Akella and M.H. Conklin, Gap-filling snow-depth time-series with Kalman filtering-smoothing and expectation maximization: Proof of concept using spatially dense wireless-sensor-network data, *Cold Regions Science and Technology* **175** (2020), 103066.
[39] S. Tavakoli, H. Fasih, J. Sadeghi and H. Torabi, Kalman Filter-Smoothed Random Walk Based Centralized Controller for Multi-Input Multi-Output Processes, *International Journal of Industrial Electronics, Control and Optimization* **2**(2) (2019), 155–166.
[40] S. Sarkka, A. Vehtari and J. Lampinen, Time series prediction by Kalman smoother with cross-validated noise density, in: *2004 IEEE International Joint Conference on Neural Networks (IEEE Cat. No. 04CH37541)*, Vol. 2, IEEE, 2004, pp. 1653–1657.
[41] T.C. Mills and T.C. Mills, *Time series techniques for economists*, Cambridge University Press, 1991.
[42] G.E. Box, G.M. Jenkins, G.C. Reinsel and G.M. Ljung, *Time series analysis: forecasting and control*, John Wiley & Sons, 2015.
[43] L. Breiman and A. Cutler, Random Forests http://www. stat. berkeley. edu/users/breiman, *RandomForests/cc_home. htm (diakses tgl 27 Juli 2016)* (2005).
[44] C.M. Bishop, *Pattern recognition and machine learning*, Springer, 2006.
[45] N.J. Nagelkerke et al., A note on a general definition of the coefficient of determination, *Biometrika* **78**(3) (1991), 691–692.
[46] T. Chai and R.R. Draxler, Root mean square error (RMSE) or mean absolute error (MAE), *Geoscientific Model Development Discussions* **7**(1) (2014), 1525–1534.

[47] M. Friedman, A comparison of alternative tests of significance for the problem of m rankings, *The Annals of Mathematical Statistics* **11**(1) (1940), 86–92.
[48] O.J. Dunn, Multiple comparisons using rank sums, *Technometrics* **6**(3) (1964), 241–252.