Sistema de predicción de variables meteorológicas utilizando Machine Learning y Software Libre

Segovia Tapia, Jenny Aracely y Toaquiza Camalle, Jonathan Fernando

Departamento de Eléctrica y Electrónica

Carrera de Ingeniería en Electrónica e Instrumentación

Artículo académico, previo a la obtención del título de Ingeniero en Electrónica e Instrumentación

Directora: Ing. Llanos Proaño, Jacqueline del Rosario Ph.D

Co - Director: Rivas Lalaleo, David Raimundo Ph.D

13 de febrero del 2023

Latacunga

electronics

MDPI

Article

Meteorological Variables Forecasting System Using Machine Learning and Open-Source Software

Jenny Aracely Segovia *, Jonathan Fernando Toaquiza *, Jacqueline Rosario Llanos * and David Raimundo Rivas

Department of Electrical and Electronic Engineering, Universidad de las Fuerzas Armadas (ESPE), Sangolquí 171103, Ecuador; drrivas@espe.edu.ec

* Correspondence: jasegovia4@espe.edu.ec (J.A.S.); tcjonathan@espe.edu.ec (J.F.T.); jdllanos1@espe.edu.ec (J.R.L)

Abstract: The techniques for forecasting meteorological variables are highly studied since prior knowledge of them allows for the efficient management of renewable energies, and also for other applications of science such as agriculture, health, engineering, energy, etc. In this research, the design, implementation, and comparison of forecasting models for meteorological variables have been performed using different Machine Learning techniques as part of Python open-source software. The techniques implemented include multiple linear regression, polynomial regression, random forest, decision tree, XGBoost, and multilayer perceptron neural network (MLP). To identify the best technique, the mean square error (RMSE), mean absolute percentage error (MAPE), mean absolute error (MAE), and coefficient of determination (R^2) are used as evaluation metrics. The most efficient techniques depend on the variable to be forecasting, however, it is noted that for most of them, random forest and XGBoost techniques present better performance. For temperature, the best performing technique was Random Forest with an R^2 of 0.8631, MAE of 0.4728 °C, MAPE of 2.73%, and RMSE of 0.6621 °C; for relative humidity, was Random Forest with an R^2 of 0.8583, MAE of 2.1380RH, MAPE of 2.50 % and RMSE of 2.9003 RH; for solar radiation, was Random Forest with an R^2 of 0.7333, MAE of 65.8105 W/m^2, and RMSE of 105.9141 W/m^2; and for wind speed, was Random Forest with an R^2 of 0.3660, MAE of 0.1097 m/s, and RMSE of 0.2136 m/s.

Keywords: machine learning; forecasting models; meteorological variables; Python

Citation: Segovia, J.A.; Toaquiza, J.F.; Llanos, J.R.; Rivas, D.R. Meteorological Variables Forecasting System Using Machine Learning and Open-Source Software. *Electronics* **2023**, *12*, x. https://doi.org/10.3390/xxxxx

Academic Editor: Grzegorz Dudek

Received: 4 January 2023
Revised: 4 February 2023
Accepted: 6 February 2023
Published: date

1. Introduction

Since the 17th century, meteorological variables have been of great interest throughout history, with the creation of the first instruments for measuring meteorological variables aiming to accurately predict the weather. For this purpose, mathematical and statistical methods and computer programs are used, most of which are of a non-linear nature [1]. Nowadays, climatic conditions change under various influences. For example, atmospheric pollution is increasing, so climate change is occurring and threatening the planet [2], which is why the measurement of meteorological variables has grown in importance as the information provided by the meteorological stations is important for monitoring climate change [3].

Climate is defined by the grouping of meteorological phenomena that are related to each other; although each of them is studied separately, it must be taken into account that a change in one produces a variation in the others [4]. The actual weather is characterised by the wind, temperature, and humidity variables forced by radiative fluxes and surface latent and sensible heat fluxes. The local climate usually denotes the mean state of the atmosphere over a 20–30-year period for a given location and day (or season) of the year. For this reason, meteorological variables are usually modeled by means of computational, numerical, and statistical techniques, most of which are nonlinear [5]. Forecasting certain climatic variables is a great challenge due to the variable behavior of the climate, which

makes it impossible to optimally manage renewable energies and obtain a greater benefit from them.

There are multiple scientific studies of modeling and prediction in order to forecast future conditions of phenomena in various fields; among the most prominent are ARIMA, Chaos Theory, and Neural Networks [6]. Forecasting models have evolved in recent decades, from smart systems with formal rules and logical theories, to the emergence of artificial intelligence techniques that allow us to propose alternatives in the treatment of information [7].

Currently, forecasting models have a high impact and are used for several applications, such as management of energy units for renewable resources microgrids [8,9], load estimation methods for isolated communities that do not receive energy or only receive it for a limited time each day [10,11], the operation of energy systems [12,13], in agriculture to predict the water consumption of plants and plan the irrigation sheet [14], in agriculture 4.0 for the prediction of variables that affect the quality of crops, for micronutrient analysis and prediction of soil chemical parameters [15], optimization of agricultural procedures and increasing productivity in the field, forecasting of SPI and Meteorological Drought Based on the Artificial Neural Network and M5P Model Tree [16], and in controllers based on forecasting models and predictive controllers. They are also used in the health field to predict the solar radiation index and to obtain a correct assessment in people with skin cancer [17], therefore, all the applications mentioned above need forecasting models that have the lowest error rate for their effective operation.

Having a forecasting model system is costly because computer packages are used in which licensing costs can be significant. On the other hand, free software is an option to reduce costs. This research proposes a system based on free software (Python), which is currently used at industrial level for its reliability, for example in applications such as the following: Advanced Time Series: Application of Neural Networks for Time Series Forecasting [18], Machine Learning in Python: main developments and technological trends in data science, Machine Learning and artificial intelligence [19], Development of an smart tool focused on artificial vision and neural networks for weed recognition in rice plantations, using Python programming language [20], etc.

In this research, different prediction techniques were evaluated and compared—among them, multiple linear regression, polynomial regression, random forest, decision tree, XGBoost, and multilayer perceptron neural network—in order to identify the best performing strategy, using evaluation metrics such as the root mean square error (RMSE) and the coefficient of determination (R^2).The variables to be predicted are temperature, relative humidity, solar radiation, and wind speed, from data taken from the weather station located in Ecuador, Tungurahua province, Baños. The predicted variables will be the inputs for a smart irrigation system and used for an energy management system of a microgrid based on predictive control, therefore, models with high approximation to online measurements are required.

The contributions of this work are as follows: (i) To design, validate, and compare different machine learning techniques, and with them select the best technique that adapts to climate variables for agriculture and energy applications, (ii) To develop a forecast system for climate variables of low cost based in free software (Python), (iii) To generate forecasting models that can be replicated for other types of variables applied to smart control systems based on forecasting models.

2. Design of Forecasting Models for Meteorological Variables

This section describes the prediction techniques used and their design. In this research, the following meteorological variables are studied and predicted: temperature, relative humidity, wind speed, and solar radiation.

The techniques designed, evaluated, and compared are the following: multiple linear regression, polynomial regression, random forest, decision tree, XGBoost, and neural

network—multilayer perceptron. To obtain the forecast of meteorological variables, the design methodology shown in Figure 1 is implemented.

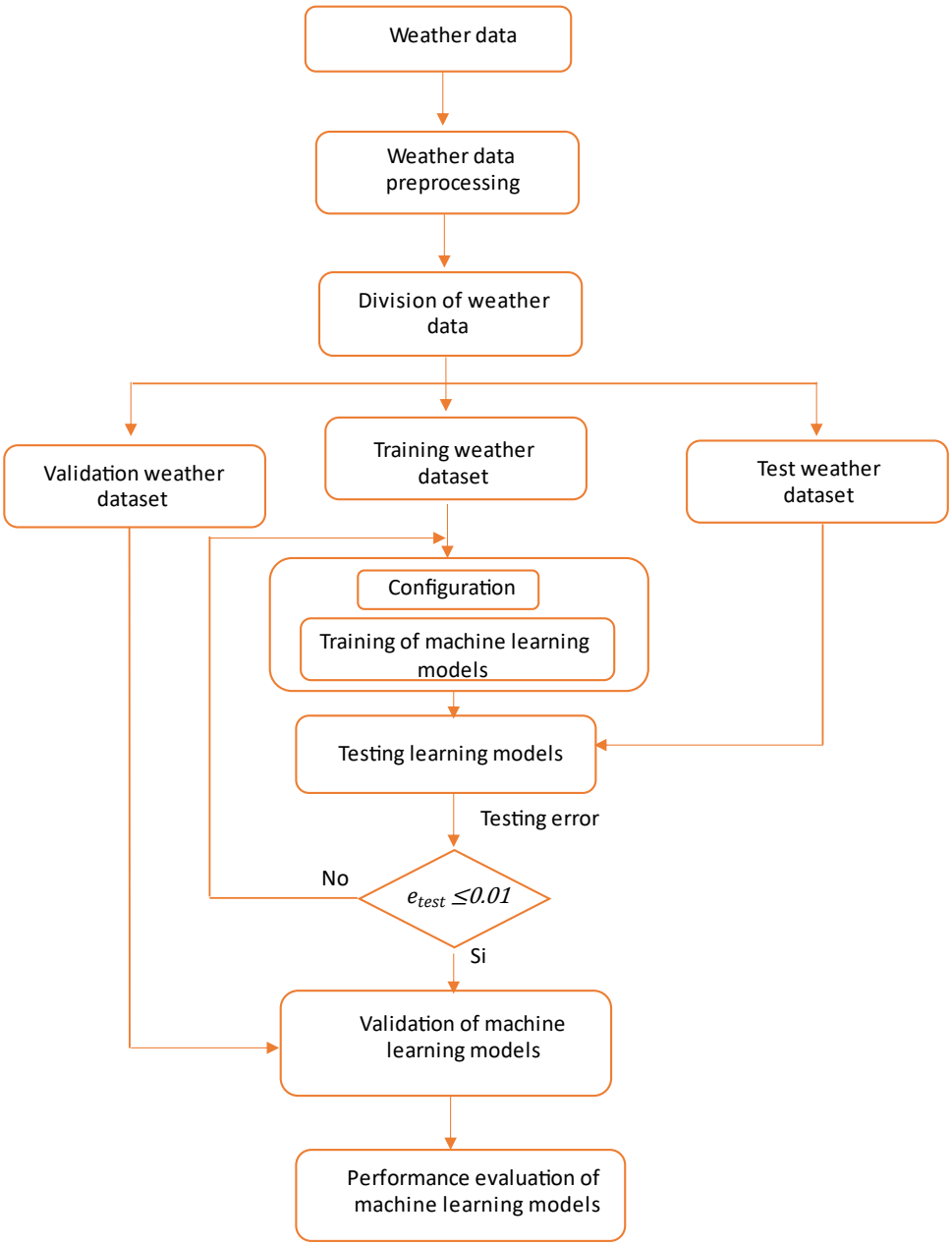


Figure 1. Flowchart of the methodology used to obtain forecasting models for meteorological variables.

2.1. Obtaining the Database

For the implementation of the forecasting models, information was obtained from the page of the Tungurahua hydrometeorological network, where there are several meteorological stations, including the Baños family park, located in Ecuador, Tungurahua province, Baños, coordinates $X = 9845439, Y = 791471$ that counts the parameters of precipitation (mm), temperature (°C), relative humidity (%), wind speed (m/s), wind direction (°), solar radiation (W/m^2), and evapotranspiration (mm). For the design of the models, only the values of temperature, solar radiation, relative humidity, and wind speed were taken, since after a previous analysis of correlation between meteorological variables, the variables with lower correlation with the variable to be predicted are

discarded. It is important to note that the values of temperature, solar radiation (net solar radiation at surface), and relative humidity were measured at a distance of 2 m, while the wind speed was measured at 10 m.

2.2. Data Preprocessing

From the database obtained, 1 year of information was available (from 23 July 2021 to 15 June 2022), which was preprocessed to take data every 5 min for each variable (temperature, relative humidity, wind speed, and solar radiation). To make a forecast, it is important to verify that there are no missing data in the measurements or to implement a data filling method; in this case, a Python algorithm was implemented, which calculates the average of the existing list of data and automatically fills in the missing data.

2.3. Dataset Division

To verify that the models work correctly, the available database is divided into three groups: training set, test set, and validation set. As its name indicates, the first one will be used to train the forecasting models, the second one will be used to evaluate the test set, and the third one to validate each of the implemented models [17,21].

After data preprocessing, a total of 93,780 data were obtained for each variable, where 80% of the database (75,024 data) is used to train the models, 20% (18,756 data) to test the models, and 2 days (576 data) were used for the validation of the models.

2.4. Design of the Forecasting Models

2.4.1. Multiple Linear Regression

It is a technique that allows modeling the relationship between a continuous variable and one or more independent variables by adjusting a linear equation. It is called simple linear regression when there is one independent variable, and if there is more than one, it is called multiple linear regression. In this context, the modeled variables are called dependent or response variables (y); and the independent variables are called regressors, predictors, or features (X) [22]. Multiple linear regression is defined by Equation (1)

$$y = a + b_1X_1 + b_2X_2 + \cdots + b_nX_n \quad (1)$$

where: $X_1, X_2, \ldots X_n$: are the predictor or independent variables, $b_1, b_2, \ldots b_n$: coefficients of the predictor variables, a: constant of the relationship between the dependent and independent variable, and y: predicted or dependent variable.

After performing different heuristic tests and using sensitivity analysis for this forecasting technique, it is deduced that the best parameters for tuning are those described in Table 1.

Table 1. Tuning parameters for the multiple linear regression techniques.

Multiple Linear Regression	
Predicted Variable	**Inputs Variables**
Temperature	Solar radiation, relative humidity, wind speed
Solar radiation	Temperature, relative humidity, wind speed
Wind speed	Temperature, solar radiation, relative humidity
Relative Humidity	Temperature, solar radiation, wind speed

2.4.2. Polynomial Regression

A linear regression with polynomial attributes that uses the relationship between the dependent (y) and independent (X) variables to find the best way to draw a line through the data points. This technique is used when the data are more complex than a simple straight line [23], and is defined by Equation (2).

$$y = a + b_1 X_i + b_2 {X_i}^2 + b_3 {X_i}^3 + \cdots + b_n {X_i}^n \quad (2)$$

where: $X_1, X_2, \ldots X_n$: are the predictor or independent variables, $b_1, b_2, \ldots b_n$: coefficients of the predictor variables, a: constant of the relationship between the dependent and independent variable, and y: predicted or dependent variable.

After performing different heuristic tests and using sensitivity analysis for this forecasting technique, it is deduced that the best parameters for tuning are those described in Table 2.

Table 2. Tuning parameters for polynomial regression technique.

Polynomial Regression		
Predicted Variable	**Inputs Variables**	**Degree of the Polynomial**
Temperature	Solar radiation, relative humidity, wind speed	4
Solar radiation	Temperature, relative humidity, wind speed	5
Wind speed	Temperature, solar radiation, relative humidity	6
Relative Humidity	Temperature, solar radiation, wind speed	4

2.4.3. Decision Tree

Values by learning decision rules derived from features and can be used for classification, regression, and multi-output tasks. Decision trees work by dividing the feature space into several simple rectangular regions, divided by parallel divisions of axes. To obtain a prediction, the mean or mode of the responses of the training observations, within the partition to which the new observation belongs, is used [23]. This is defined by Equation (3).

$$G_i = 1 - \sum_{k=1}^{m} (P_{i,k})^2 \quad (3)$$

where: $P_{i,k}$: is the radio of class k instances among the training instances in the i^{th} node, m: number of class labels, and G_i (Gini impurity): represents the measure for constructing decision trees.

After performing different heuristic tests and using sensitivity analysis for this forecast technique, it is deduced that the best parameters for tuning are those described in Table 3.

Table 3. Tuning parameters for the decision tree technique.

Decision Tree			
Predicted Variable	**Inputs Variables**	**Max_Depth**	**Min_Samples_Leaf**
Temperature	Solar radiation, relative humidity, wind speed	10	18
Solar radiation	Temperature, relative humidity, wind speed	10	7
Wind speed	Temperature, solar radiation, relative humidity	19	6
Relative Humidity	Temperature, solar radiation, wind speed	9	16

2.4.4. Random Forest

A supervised learning algorithm that uses an ensemble learning method for regression that combines predictions from several machine learning algorithms (decision trees) to make a more accurate prediction than a single model [23]. Figure 2 shows that the random forest algorithm is composed of a collection of decision trees, and each tree in the set is composed of a sample of data extracted from a training set (DATASET); for a regression task, the individual decision trees are averaged (Average) until the predicted value (Prediction) is obtained.

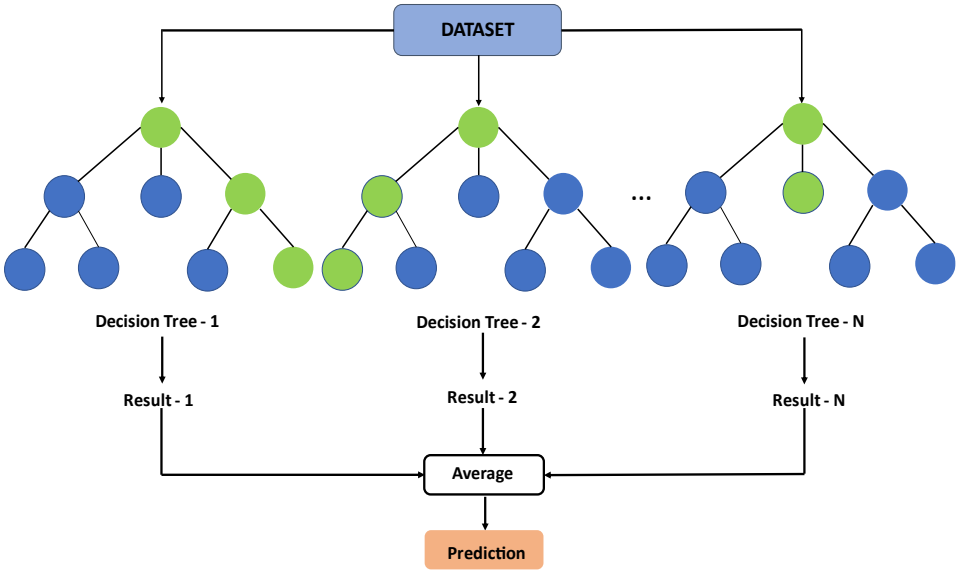


Figure 2. Algorithm for making predictions using random forest.

In general, deep decision trees tend to overfit, while random forests avoid this by generating random subsets of features and using those subsets to build smaller trees. The generalization error for random forests is based on the strength of the individual constructed trees and their correlation [24].

This technique has several parameters that can be configured, such as the following:

N° estimators: the number of trees in the forest. Max leaf nodes: the maximum number of leaf nodes, this hyperparameter sets a condition for splitting the tree nodes and thus restricts the growth of the tree. If after splitting there are more terminal nodes than the specified number, the splitting stops and the tree does not continue to grow, which helps to avoid overfitting. And Max features: the maximum number of features that are evaluated for splitting at each node, increasing max_features generally improves model performance, since each node now has a greater number of options to consider [23].

After performing different heuristic tests and using sensitivity analysis for this forecast technique, it is deduced that the best parameters for tuning are those described in Table 4.

Table 4. Tuning parameters for the random forest technique.

Random Forest				
Predicted Variable	**Inputs Variables**	**N° Estimators**	**Max Leaf Nodes**	**Max Features**
Temperature	Solar radiation, relative humidity, wind speed	100	3000	0.1
Solar radiation	Temperature, relative humidity, wind speed	100	3000	0.1
Wind speed	Temperature, solar radiation, relative humidity	100	2000	0.3

Relative Humidity	Temperature, solar radiation, wind speed	100	2000	0.2

2.4.5. Extreme Gradient Boosting (XGboost)

The XGBoost algorithm is a scalable tree-boosting system that can be used for both classification and regression tasks. It performs a second-order Taylor expansion on the loss function and can automatically use multiple threads of the central processing unit (CPU) for parallel computing. In addition, XGBoost uses a variety of methods to avoid overfitting [25].

Figure 3 shows the XGBoost algorithm; decision trees are created sequentially (Decision Tree-1, Decision Tree-2, Decision Tree-N) and weights play an important role in XGBoost. Weights are assigned to all independent variables, which are then entered into the decision tree that predicts the outcomes (Result -1, Result-2, Result -N). The weights of variables incorrectly predicted by the tree are increased and these variables are then fed into the second decision tree (Residual error). These individual predictors are then grouped (Average) to give a strong and more accurate model (Prediction).

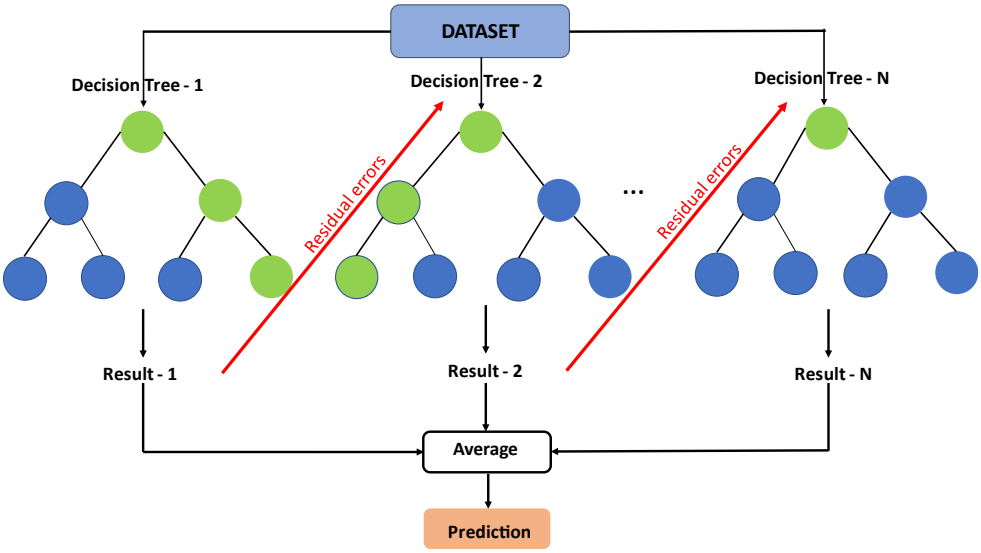


Figure 3. Structure of an XGBoost algorithm for regression.

After performing different heuristic tests and using sensitivity analysis for this forecast technique, it is deduced that the best parameters for its tuning are those described in Table 5.

Table 5. Tuning parameters for the XGboost technique.

XGBoost			
Predicted Variable	**Inputs Variables**	**Max Depth**	**N° Estimators**
Temperature	Solar radiation, relative humidity, wind speed	2	100
Solar radiation	Temperature, relative humidity, wind speed	2	20
Wind speed	Temperature, solar radiation, relative humidity	5	19
Relative Humidity	Temperature, solar radiation, wind speed	7	19

2.4.6. Neural Network—Multilayer Perceptron

It is an effective and widely used model for modeling many real situations. The multilayer perceptron is a hierarchical structure consisting of several layers of fully interconnected neurons, which input neurons are outputs of the previous layer. Figure 4 shows the structure of a multilayer perceptron neural network; the input layer is made up of r units (where r is the number of external inputs) that merely distribute the input signals to the next layer; the hidden layer is made up of neurons that have no physical contact with the outside; the number of hidden layers is variable (u); and the output layer is made up of l neurons (where l is the number of external outputs) whose outputs constitute the vector of external outputs of the multilayer perceptron [26].

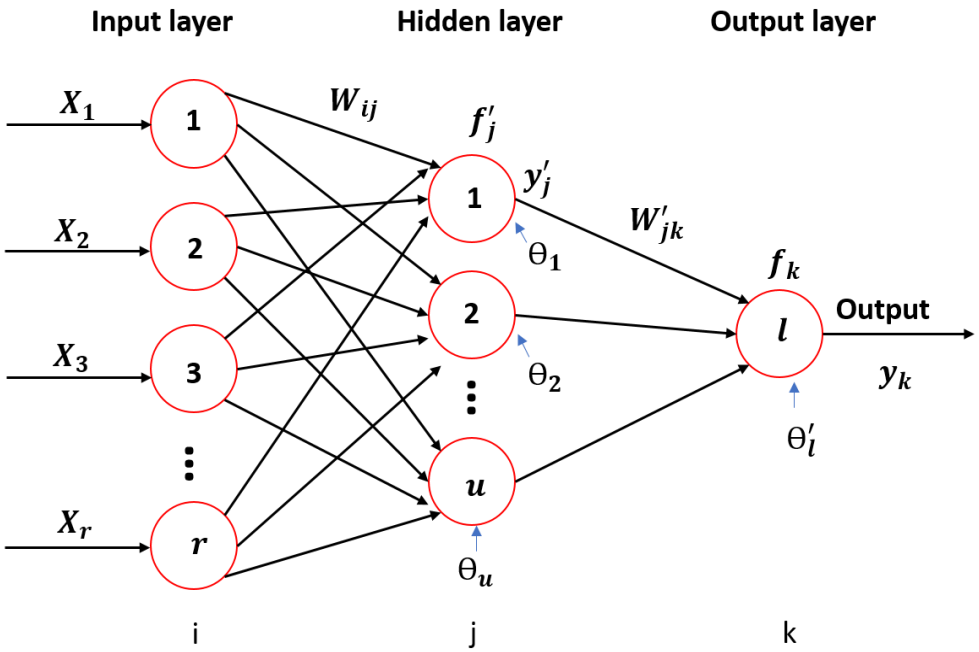


Figure 4. Structure of a multilayer perceptron neural network.

The training of the neural network consists of calculating the linear combination from a set of input variables, with a bias term, applying an activation function, generally the threshold or sign function, giving rise to the network output. Thus, the weights of the network are adjusted by the method of supervised learning by error correction (backpropagation), in such a way that the expected output is compared with the value of the output variable to be obtained, the difference being the error or residual. Each neuron behaves independently of the others: each neuron receives a set of input values (an input vector), calculates the scalar product of this vector and the vector of weights, adds its own bias, applies an activation function to the result, and returns the final result obtained [26].

In general, all weights and biases will be different. The output of the multilayer perceptron neural network is defined by equation (4). Where: y_k is the output, f_k activation function of output layer, θ'_k bias of the output layer, W_{ij} hidden layer weights, y'_joutput of the hidden layer, f'_j activation function of the hidden layer, X_i neuron inputs, W'_{jk} output layer weights, θ_j bias of hidden layer, r is the number of inputs for the neuron j from the hidden layer, and u is the number of inputs for the neuron k from the output layer [27].

$$y'_j = f'_j\left(\sum_{i=1}^{r} X_i W_{ij} - \theta_j\right)$$

$$y_k = f_k\left(\sum_{j=1}^{u} y_j' W_{jk}' - \theta_k'\right) \quad (4)$$

For this research, backpropagation was used as a training technique. After performing different heuristic tests and using sensitivity analysis for this forecasting technique, it is deduced that the best parameters for its tuning are those described in Table 6.

Table 6. Tuning parameters for the multilayer perceptron neural network technique.

Neural Network—Multilayer Perceptron						
Predicted Variable	**Inputs Variables**	**Input Layer Neurons**	**N° Epoch**	**Batch Size**	**Hidden Layer Neurons**	**Activation Function**
Temperature	Solar radiation, relative humidity, wind speed	3	5000	128	32	Hidden: ReLU Out: Sigmoid
Solar radiation	Temperature, relative humidity, wind speed	3	5000	128	32	Hidden: ReLU Out: Sigmoid
Wind speed	Temperature, solar radiation, relative humidity	3	3000	128	32	Hidden: ReLU Out: Sigmoid
Relative Humidity	Temperature, solar radiation, wind speed	3	5000	128	32	Hidden: ReLU Out: Sigmoid

3. Results

3.1. Indicators for Assessing the Performance of Weather Forecasting Models

To measure the performance of the forecast techniques for each of the variables described above, two types of metrics were used: to evaluate the forecast accuracy, the mean square error RMSE is used, which allows comparing their results and defining the technique with the lowest error, and therefore, the best method for each variable to be predicted. In addition, to determine if the implemented models perform well in their training and to define their predictive ability, the coefficient of determination is R^2.

3.1.1. Coefficient of determination (R^2)

R^2 or coefficient of determination can be in the range of $[-\infty, 1]$ it is used to determine the ability of a model to predict future results. The best possible result is 1, and occurs when the prediction coincides with the values of the target variable, while the closer to zero, the less well-fitted the model is and, therefore, the less reliable it is. R^2 can take negative values because the prediction can be arbitrarily bad [28]. It is defined as equation (5), described by 1 minus the sum of total squares divided by the sum of squares of the residuals.

$$R^2 = 1 - \frac{\sum(y_c - \hat{y}_c)^2}{\sum(y_c - \bar{y})^2} \quad (5)$$

where: y_c: are the values taken by the target variable, $\hat{y}_c$: are the values of the prediction, and $\bar{y}$: is the mean value of the values taken by the target variable.

3.1.2. Mean square error (RMSE)

The root mean square error, also known as root mean square deviation, measures the amount of error between two sets of data. That is, it compares the predicted value with the observed or known value [28]. It is given by Equation (6):

$$RMSE = \sqrt{\frac{1}{o}\sum_{c=1}^{o}(y_c - \hat{y}_c)^2} \quad (6)$$

where: y_c: are the values taken by the target variable, $\hat{y}_c$: are the values of the prediction, and o: is the sample size.

3.1.3. Mean Absolute Percentage Error (MAPE)

Mean absolute percentage error is an evaluation metric for regression problems; the idea of this metric is to be sensitive to relative errors. MAPE is the mean of all absolute percentage errors between the predicted and actual values [29]. It is given by Equation (7):

$$MAPE = \frac{1}{o}\sum_{c=1}^{o}\left|\frac{y_c - \hat{y}_c}{y_c}\right| * 100\% \quad (7)$$

where y_c: are the values taken by the target variable, $\hat{y}_c$: are the values of the prediction, and o: is the sample size.

Equation (7) helps to understand one of the important caveats when using MAPE, since to calculate this metric, you need to divide the difference by the actual value. This means that if you have actual values close to 0 or at 0, the MAPE score will receive a division error by 0 or will be extremely high. Therefore, it is recommended not to use MAPE when it has real values close to 0 [30].

3.1.4. Mean Absolute Error (MAE)

Mean absolute error is a common metric to use for measuring the error of regression predictions. The mean absolute error of a model is the mean of the absolute values of the individual prediction errors on over all instances in the test set. Each prediction error is the difference between the true value and the predicted value for the instance [16,31]. It is given by Equation (8):

$$MAE = \frac{1}{o}\sum_{c=1}^{o}|\, y_c - \hat{y}_c| \quad (8)$$

where: y_c: are the values taken by the target variable, $\hat{y}_c$: are the values of the prediction, and o: is the sample size.

3.2. Case Study

For the implementation of the forecast techniques for meteorological variables (temperature, wind speed, solar radiation, and relative humidity), the Python programming language was used. Information was obtained from the Parque de la Familia Baños meteorological station, located in Ecuador, Tungurahua province, Baños, coordinates $X = 9845439$, $Y = 791471$. From the database obtained, 1 year of information was available (from 23 July 2021 to 15 June 2022) with a sampling time of 5 min having a total of 93,780 data for each variable, where 80% of the database (75,024 data) is used to test the models, 20% (18,756 data) to test the models, and 2 days (576 data) were used for validation. To obtain the values of the evaluation metrics (RMSE, MAE, MAPE y R^2) the validation data corresponding to the days 10/06/2022 and 11/06/2022 were used.

The forecast techniques implemented for all variables are the following: multiple linear regression, polynomial regression, decision tree, random forest, XGboost, and multilayer perceptron neural network.

To identify which of the models is more efficient, evaluation metrics such as root mean square error (RMSE), mean absolute percentage error (MAPE), and mean absolute error (MAE) are used over the entire validation range, while to evaluate whether the forecasting algorithms fit correctly, the R^2 metric is used. It is important to note that these

metrics evaluate different aspects; the RMSE, MAPE, and MAE evaluate the forecasting error, while R^2 allows to analyze how well a regression model fits the real data.

3.2.1. Temperature Forecasting

Table 7 shows the results of the evaluation metrics: root mean square error (RMSE), mean absolute percentage error (MAPE), mean absolute error (MAE), and coefficient of determination (R^2) for each of the techniques used for temperature forecasting. The calculation of the root mean square error, mean absolute percentage error, and mean absolute error was obtained by averaging the errors of the validation data (576 data), while the calculation of the coefficient of determination (R^2) used the data from the training set and the test set (93,780 data).

Table 7 shows that R^2 obtained from the implemented algorithms converge to appropriate values, i.e., there is a correct approximation between the real temperature and the predicted temperature, thus guaranteeing the good performance of the algorithm, which allows a comparison of the performance in terms of forecast error. Comparison of the root mean square errors (RMSE), mean absolute percentage errors (MAPE), and mean absolute errors (MAE), and analysis of the coefficient of determination R^2 of the different techniques implemented show that the best performing technique for forecasting the temperature variable is Random Forest, with an R^2 of 0.8631, MAE of 0.4728 °C, MAPE of 2.73%, and RMSE of 0.6621 °C. This is followed by XGBoost, with an R^2 of 0.8599, MAE of 0.5335 °C, MAPE of 3.09%, and RMSE of 0.7565 °C.

Figure 5 shows the real (red) and prediction (blue) profiles using the different Machine Learning techniques to predict the temperature variable: (a) Multiple linear regression technique, (b) Polynomial regression technique, (c) Decision tree technique, (d) Random Forest technique, (e) XGboost technique, (f) Multilayer perceptron neural network technique. Figure 5c,d, validate that the best performance corresponds to the Decision tree and Random forest techniques.

Table 7. Evaluation metrics for temperature forecasting.

Technique	Coefficient of Determination (R^2)	Mean Absolute Error (MAE) [°C]	Mean Absolute Percentage Error (MAPE)[%]	Mean Square Error (RMSE) [°C]
Multiple linear regression	0.8244	0.6597	3.71	0.8453
Polynomial regression	0.8406	0.6097	3.51	0.8146
Decision tree	0.8593	0.5097	2.95	0.7333
Random forest	0.8631	0.4728	2.73	0.6621
XGboost	0.8599	0.5335	3.09	0.7565
Multilayer perceptron	0.8226	0.9124	5.51	1.2498

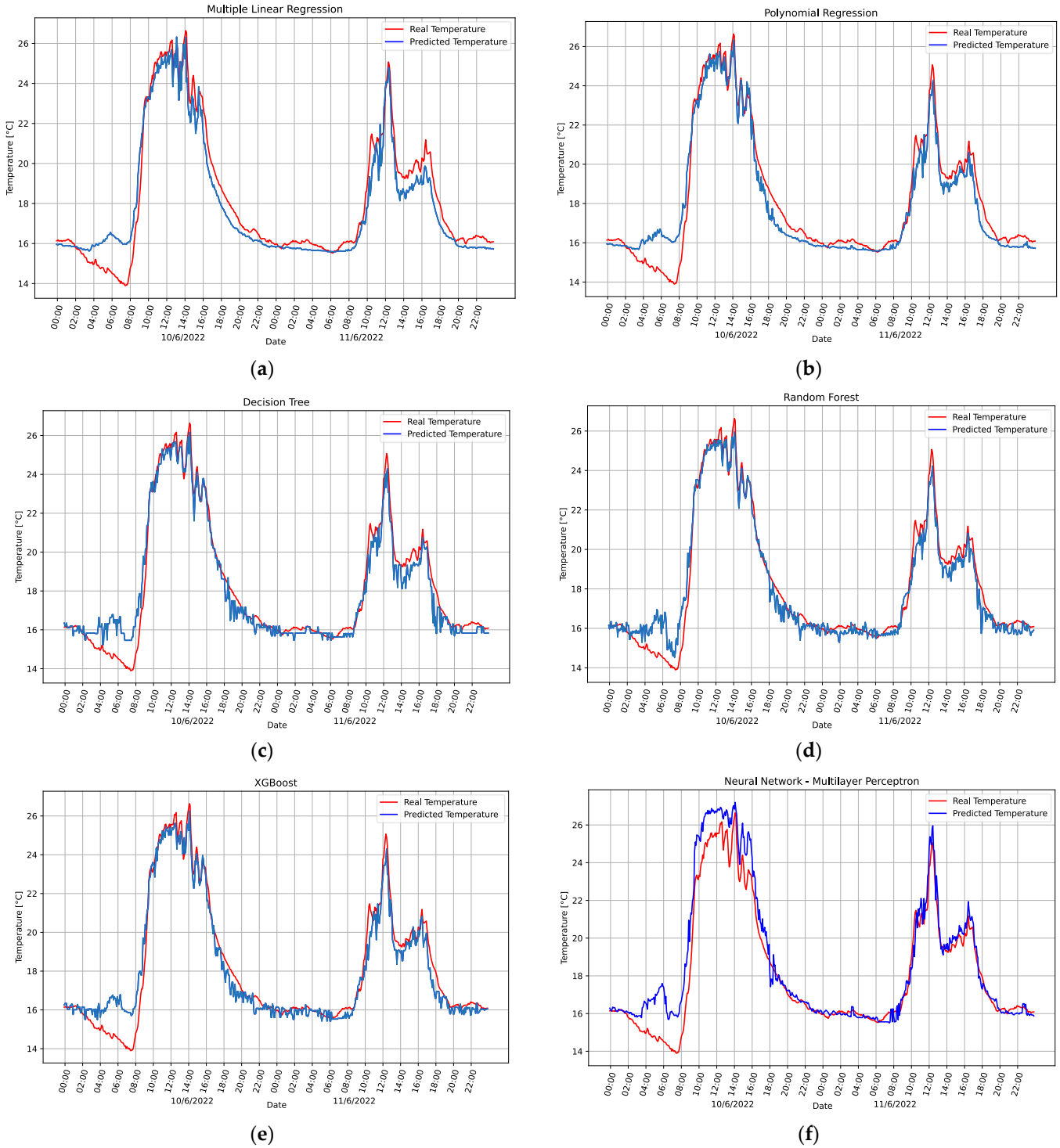


Figure 5. Temperature forecast techniques: (**a**) Multiple linear regression, (**b**) Polynomial regression, (**c**) Decision tree, (**d**) Random forest, (**e**) XGboost, (**f**) Multilayer perceptron neural network.

3.2.2. Relative Humidity Forecasting

Table 8 shows the results of the evaluation metrics: root mean square error (RMSE), mean absolute percentage error (MAPE), mean absolute error (MAE), and coefficient of determination (R^2) for each of the techniques used for relative humidity forecasting. The calculation of the root mean square error, mean absolute percentage error, and mean absolute error was obtained by averaging the errors of the validation data (576 data), while

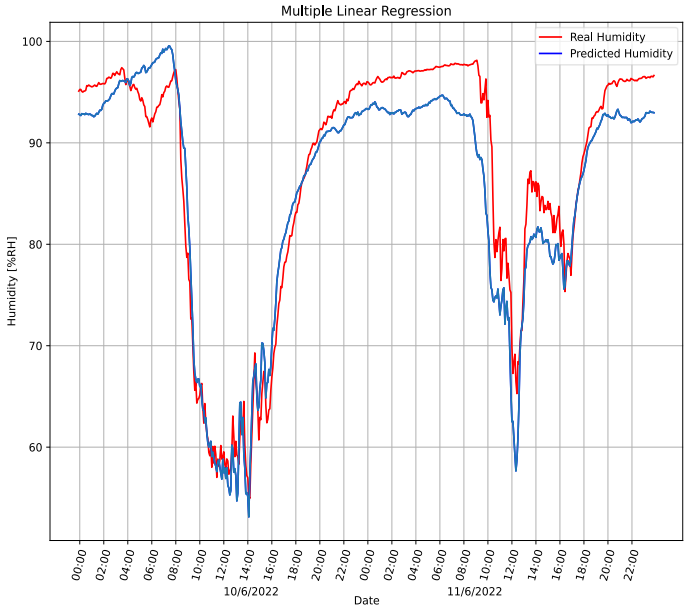
the calculation of the coefficient of determination (R^2) used the data from the training set and the test set (93,780 data).

Table 8 shows that R^2 obtained from the implemented algorithms converge to appropriate values, i.e., there is a correct approximation between the real relative humidity and the predicted relative humidity, thus guaranteeing the good performance of the algorithm, which allows a comparison of the performance in terms of forecast error. Comparison of the root mean square errors (RMSE), mean absolute percentage errors (MAPE), and mean absolute errors (MAE), and analysis of the coefficient of determination R^2 of the different techniques implemented show that the best performing techniques for forecasting the relative humidity variable are Random Forest, with an R^2 of 0.8583, MAE of 2.1380 RH, MAPE of 2.50%, and RMSE of 2.9003 RH; and XGBoost, with an R^2 of 0.8597, MAE of 2.2907 RH, MAPE of 2.67%, and RMSE of 3.1444 RH.

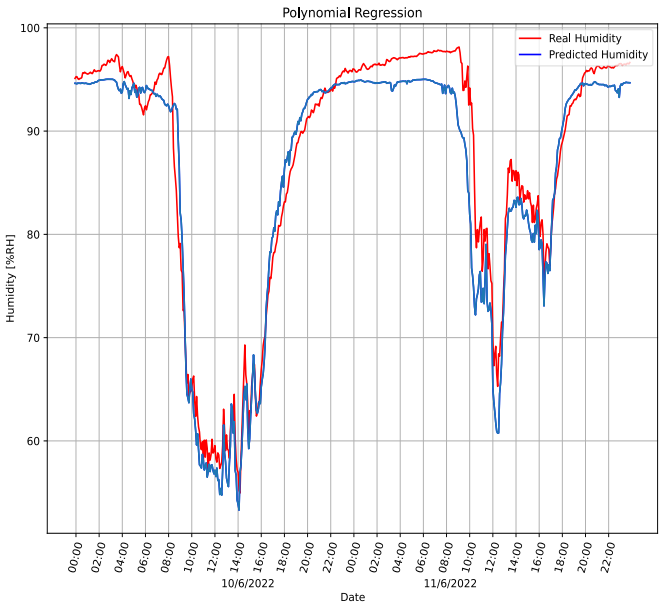
Figure 6 shows the real (red) and prediction (blue) profiles using the different Machine Learning techniques to predict the relative humidity variable: (a) Multiple linear regression technique, (b) Polynomial regression technique, (c) Decision tree technique, (d) Random forest technique, (e) XGboost technique, (f) Multilayer perceptron neural network technique. Figure 6d and Figure 6c validate that the best performance corresponds to the Random forest and Decision tree techniques.

Table 8. Evaluation metrics for relative humidity forecasting.

Technique	Coefficient of Determination (R^2)	Mean Absolute Error (MAE) [RH]	Mean Absolute Percentage Error (MAPE)[%]	Mean Square Error (RMSE) [RH]
Multiple linear regression	0.7815	3.0900	3.56	3.7475
Polynomial regression	0.8420	2.2816	2.68	3.0163
Decision tree	0.8547	2.2685	2.65	3.2083
Random forest	0.8583	2.1380	2.50	2.9003
XGboost	0.8597	2.2907	2.67	3.1444
Multilayer perceptron	0.8013	4.6055	5.64	5.5759



(**a**)



(**b**)

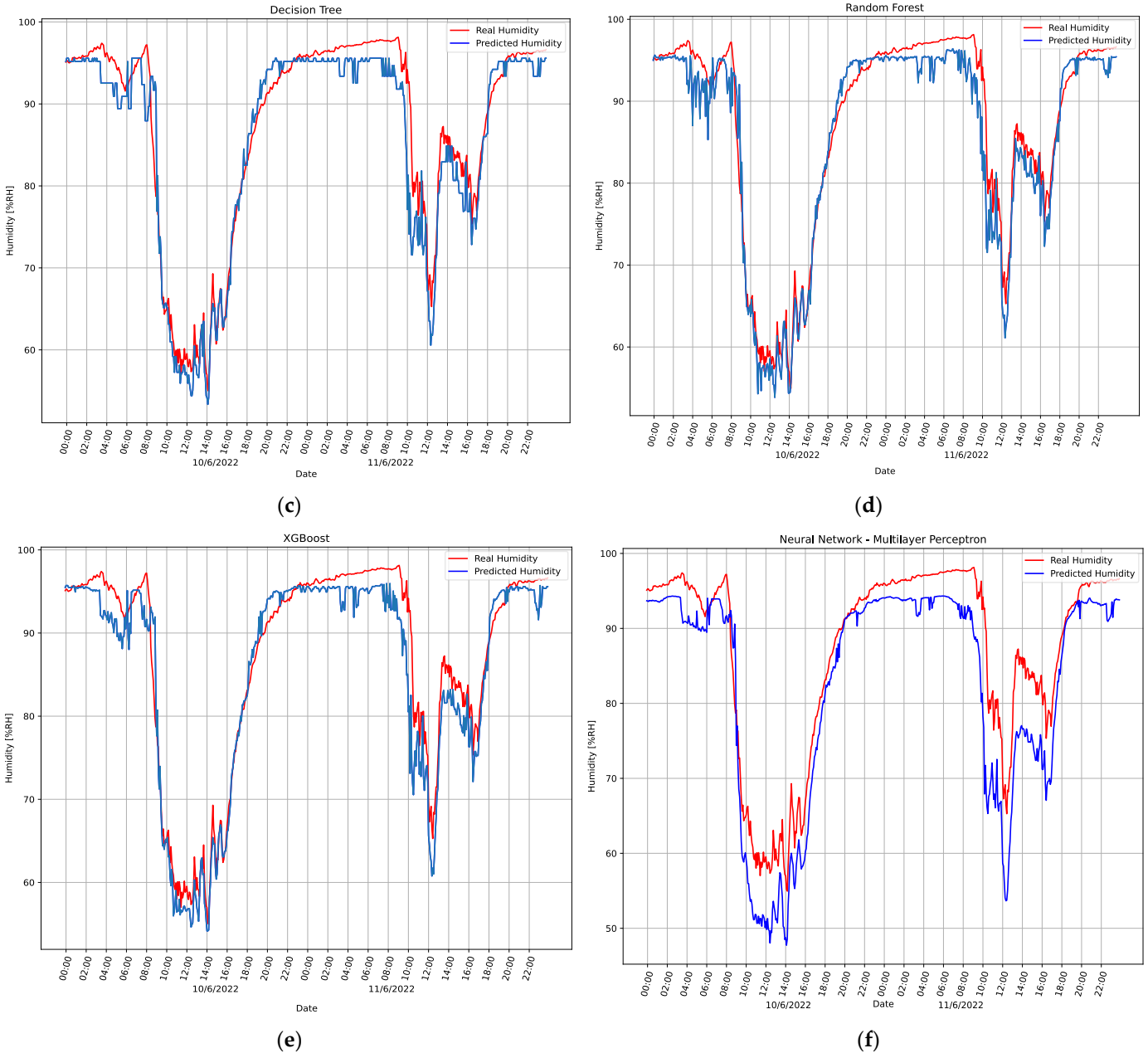


Figure 6. Techniques for relative humidity forecasting: (**a**) Multiple linear regression, (**b**) Polynomial regression, (**c**) Decision tree, (**d**) Random forest, (**e**) XGboost, (**f**) Multilayer perceptron neural network.

3.2.3. Solar Radiation Forecasting

Table 9 shows the results of the evaluation metrics: root mean square error (RMSE), mean absolute error (MAE), and coefficient of determination (R^2) for each of the techniques used for solar radiation forecasting. The calculation of the root mean square error, and mean absolute error was obtained by averaging the errors of the validation data (576 data), while the calculation of the coefficient of determination (R^2) used the data from the training set and the test set (93780 data).

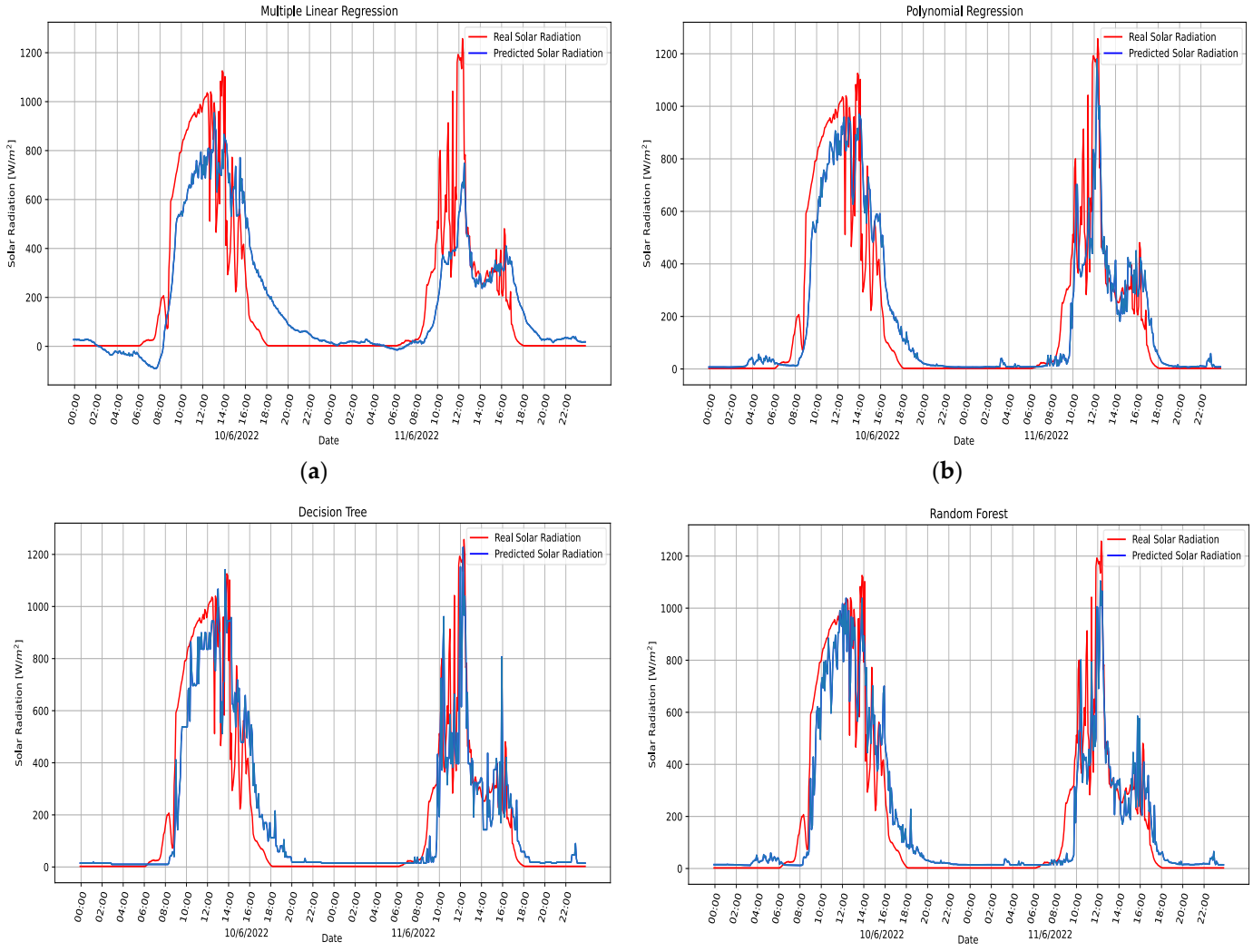
Table 9 shows that R^2 obtained from the implemented algorithms converge to appropriate values, i.e., there is a correct approximation between the real solar radiation and the predicted solar radiation, thus guaranteeing the good performance of the algorithm, which allows a comparison of the performance in terms of forecast error. Comparison of the root mean square errors (RMSE), and mean absolute errors (MAE), and analysis of the

coefficient of determination R^2 of the different techniques implemented show that the best performing techniques for forecasting the solar radiation variable are Random Forest with an R^2 of 0.7333, MAE of 65.8105 W/m^2, and RMSE of 105.9141 W/m^2 ; and Decision Tree with an R^2 of 0.7253, MAE of 75.8177 W/m^2, and RMSE of 127.3530 W/m^2.

Figure 7 shows the real (red) and prediction (blue) profiles using the different Machine Learning techniques to predict the variable solar radiation: (a) Multiple linear regression technique, (b) Polynomial regression technique, (c) Decision tree technique, (d) Random forest technique, (e) XGboost technique, (f) Multilayer perceptron neural network technique. Figure 7d validates that the best performance corresponds to the Random forest technique.

Table 9. Evaluation metrics for solar radiation forecasting.

Technique	Coefficient of Determination (R^2)	Mean Absolute Error (MAE) [W/m^2]	Mean Square Error (RMSE) [W/m^2]
Multiple linear regression	0.6689	106.9741	164.7435
Polynomial regression	0.7394	76.6667	129.1836
Decision tree	0.7253	75.8177	127.3530
Random forest	0.7333	65.8105	105.9141
XGboost	0.7075	87.6137	145.0170
Multilayer perceptron	0.7423	88.5897	140.0681



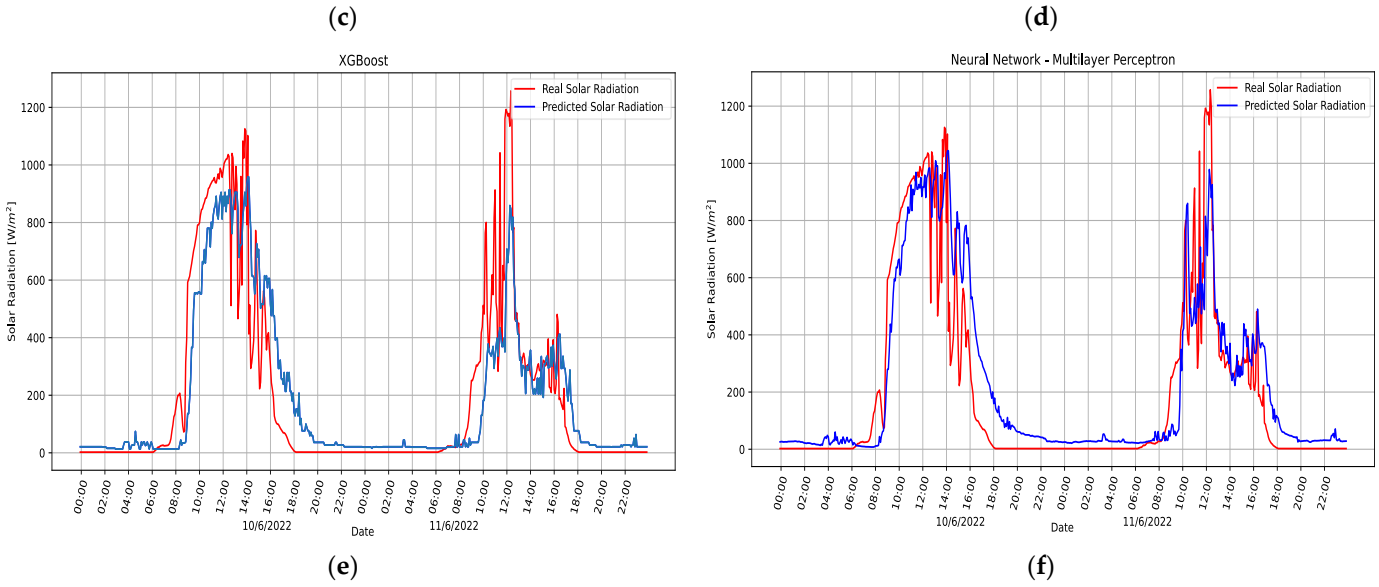


Figure 7. Solar radiation forecast techniques: (**a**) Multiple linear regression, (**b**) Polynomial regression, (**c**) Decision tree, (**d**) Random forest, (**e**) XGboost, (**f**) Multilayer perceptron neural network.

3.2.4. Wind Speed Forecasting

Table 10 shows the results of the evaluation metrics: root mean square error (RMSE), mean absolute error (MAE), and coefficient of determination (R^2) for each of the techniques used for wind speed forecasting. The calculation of the root mean square error and mean absolute error was obtained by averaging the errors of the validation data (576 data), while the calculation of the coefficient of determination (R^2) used the data from the training set and the test set (93780 data).

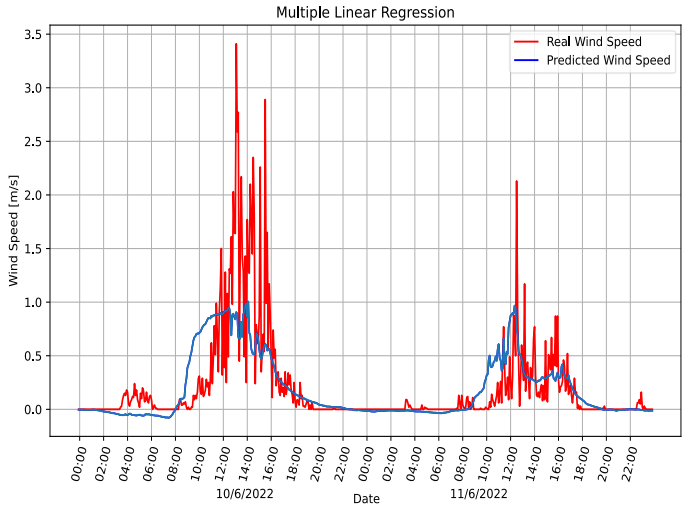
Table 10 shows that R^2 obtained from the implemented algorithms converge to appropriate values, i.e., there is an acceptable approximation between the real wind speed and the predicted wind speed, thus guaranteeing the good performance of the algorithm, which allows a comparison of the performance in terms of forecast error. Comparison of the root mean square errors (RMSE) and mean absolute errors (MAE) and analysis of the coefficient of determination R^2 of the different techniques implemented show that the best performing techniques for forecasting the wind speed variable are Random Forest with an R^2 of 0.3660, MAE of 0.1097 m/s, and RMSE of 0.2136 m/s; and XGBoost with an R^2 of 0.3866, MAE of 0.1439 m/s, and RMSE of 0.3131 m/s. It should be taken into account that due to the high variability of wind speed, the implemented techniques have a lower coefficient of determination compared to the other variables; however, forecasts with acceptable errors were obtained. In this case, the value of the mean absolute percentage errors (MAPE) is not taken into account because it is used only when it is known that the quantity to be predicted remains well above 0.

Figure 8 shows the real (red) and prediction (blue) profiles using the different Machine Learning techniques to predict the wind speed variable: (a) Multiple linear regression technique, (b) Polynomial regression technique, (c) Decision tree technique, (d) Random forest technique, (e) XGboost technique, (f) Multilayer perceptron neural network technique. Figure 8d validates that the best performance corresponds to the Random forest technique.

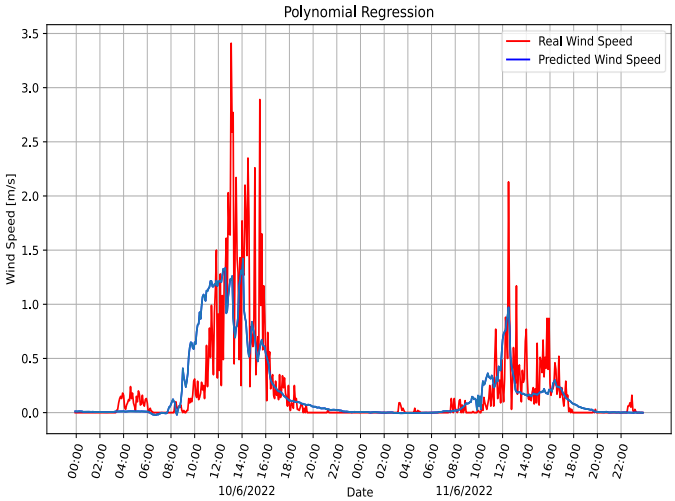
Table 10. Evaluation metrics for wind speed forecasting.

Technique	Coefficient of Determination (R^2)	Mean Absolute Error (MAE) [m/s]	Mean Square Error (RMSE) [m/s]
Multiple linear regression	0.3428	0.1614	0.3354
Polynomial regression	0.3770	0.1428	0.3159

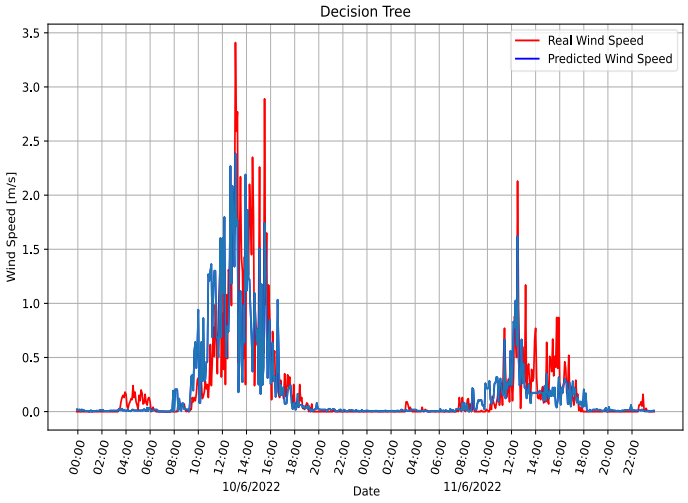
Decision tree	0.2142	0.1256	0.2705
Random forest	0.3660	0.1097	0.2136
XGboost	0.3866	0.1439	0.3131
Multilayer perceptron	0.3270	0.1654	0.3616



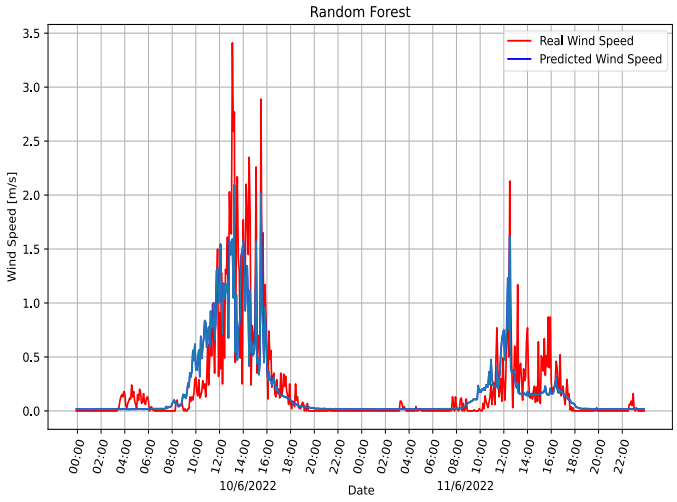
(**a**)



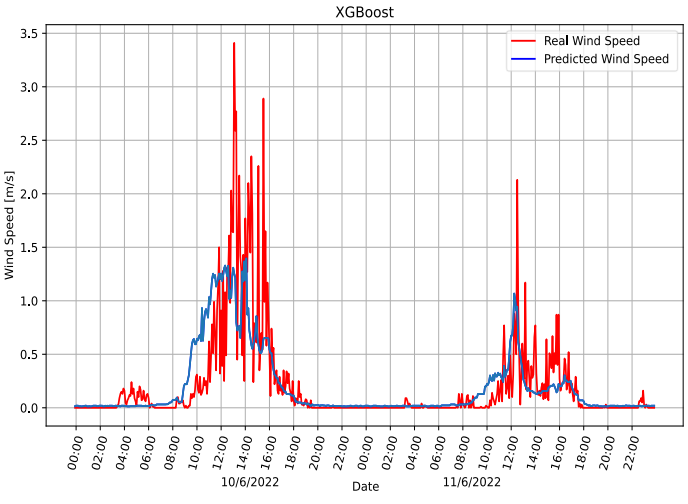
(**b**)

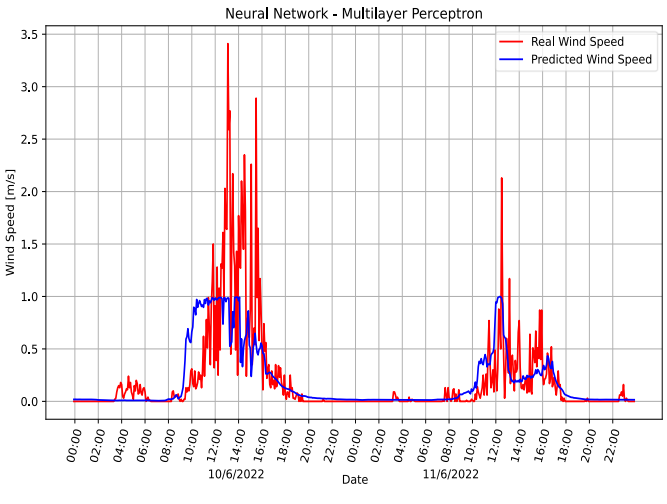


(**c**)



(**d**)





(**e**) (**f**)

Figure 8. Techniques for wind speed forecast: (**a**) Multiple linear regression, (**b**) Polynomial regression, (**c**) Decision tree, (**d**) Random forest, (**e**) XGboost, (**f**) Multilayer perceptron neural network.

4. Conclusions

For the forecasting of meteorological variables in this research, information obtained from the Parque de la Familia Baños meteorological station located in Ecuador was used and the following prediction techniques were tested: multiple linear regression, polynomial regression, decision tree, random forest, XGBoost, and multilayer perceptron neural network. For forecasting the temperature variable, a better result is obtained by using Random Forest with an R^2 of 0.8631, MAE of 0.4728 °C, MAPE of 2.73%, and RMSE of 0.6621 °C. In addition, XGBoost also performed well with an R^2 of 0.8599, MAE of 0.5335 °C, MAPE of 3.09%, and RMSE of 0.7565 °C. For forecasting the relative humidity variable, a better result is obtained by using Random Forest with an R^2 of 0.8583, MAE of 2.1380 RH, MAPE of 2.50%, and RMSE of 2.9003 RH. In addition, XGBoost also performed well with an R^2 of 0.8597, MAE of 2.2907 RH, MAPE of 2.67%, and RMSE of 3.1444 RH. For forecasting the solar radiation variable, a better result is obtained by using Random Forest with an R^2 of 0.7333, MAE of 65.8105 W/m^2, and RMSE of 105.9141 W/m^2. In addition, Decision Tree also performed well with an R^2 of 0.7253, MAE of 75.8177 W/m^2, and RMSE of 127.3530 W/m^2. For forecasting the wind speed variable, a better result is obtained by using Random Forest, with an R^2 of 0.3660, MAE of 0.1097 m/s, and RMSE of 0.2136 m/s. In addition, XGBoost also performed well, with an R^2 of 0.3866, MAE of 0.1439 m/s, and RMSE of 0.3131 m/s.

It can be observed that wind speed has the highest variability compared to the other predicted variables, therefore, the results of the techniques implemented show that the coefficient of determination R^2 of this variable has a lower value. This is due to the type of signal we are trying to predict; however, acceptable predictions were obtained.

The prediction of meteorological variables (temperature, solar radiation, wind speed, and relative humidity) will allow future projects to be implemented in the study area, such as intelligent agriculture to support food problems in that area and the implementation of a microgrid based on renewable resources where prediction models will support the planning and operation of the microgrid in real time, allowing clean energy to this locality, contributing to the reduction in the use of fossil resources, which is the goal that different countries have set as part of their policies.

Author Contributions: Conceptualization, J.A.S., J.F.T., J.R.L., and D.R.R.; methodology, , J.A.S., J.F.T., J.R.L., and D.R.R.; software J.A.S., and J.F.T.; validation, J.A.S., and J.F.T.; formal analysis, J.A.S., J.F.T., J.R.L., and D.R.R.; investigation, J.A.S., J.F.T., and J.R.L.; resources, J.A.S., and J.F.T.; data curation, J.A.S., and J.F.T.; writing—original draft preparation, J.A.S., J.F.T., J.R.L., and D.R.R..; writing—review and editing, J.A.S., J.F.T., J.R.L., and D.R.R.; visualization, J.A.S., J.F.T., J.R.L., and D.R.R.; supervision, J.R.L., and D.R.R.; project administration, J.R.L.; funding acquisition, J.R.L. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Acknowledgments: This work was supported in part by the Universidad de las Fuerzas Armadas ESPE through the Project "Optimal energy management systems for hybrid generation systems", under Project 2023-pis-03. In addition, the authors would like to thank to the project EE-GNP-0043-2021 - ESPE, REDTPI4.0 - CYTED, Conv-2022-05 - UNACH, "SISMO-ROSAS" – UPS.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest

References

1. Ayala, M.F. Analisis De La Dinamica Caoticapara La Series Temporales De Variables Meteorologicas En La Estacion Climatica De Chone, 2017. Universidad de las Fuerzas Armadas ESPE, Ecuador, 2017. Available online: http://repositorio.espe.edu.ec/handle/21000/13629 (accessed on 24 November 2022).
2. Erdil, A.; Arcaklioglu, E. The prediction of meteorological variables using artificial neural network. *Neural Comput. Appl.* **2013**, *22*, 1677–1683. https://doi.org/10.1007/s00521-012-1210-0.
3. Ruiz-Ayala, D.C.; Vides-Herrera, C.A.; Pardo-García, A. Monitoreo de variables meteorológicas a través de un sistema inalámbrico de adquisición de datos. *Rev. Investig. Desarro. e Innovación* **2018**, *8*, 333–341. https://doi.org/10.19053/20278306.v8.n2.2018.7971.
4. Inzunza, J.C. Meteorologia Descriptiva. *Univ. Concepción Dep. Geofísica* **2015**, 1–34. Available online: http://www2.udec.cl/~jinzunza/meteo/cap1.pdf (accessed on 24 November 2022).
5. Millán, H.; Kalauzi, A.; Cukic, M.; Biondi, R. Nonlinear dynamics of meteorological variables: Multifractality and chaotic invariants in daily records from Pastaza, Ecuador. *Theor. Appl. Climatol.* **2010**, *102*, 75–85. https://doi.org/10.1007/s00704-009-0242-6.
6. Acurio, W.; Pilco, V. *Técnicas estadísticas para la modelación y predicción de la temperatura y velocidad del viento en la provincia de Chimborazo*; Escuela Superior Politénica de Chimborazo; Ecuador, 2019. Available online: http://dspace.espoch.edu.ec/handle/123456789/10955 (accessed on 28 November 2022).
7. Tong, H. *Non-linear time series: A dynamical system approach*; Oxford University Press: Oxford, UK, 1990.
8. Palma-Behnke, R.; Benavides, C.; Aranda, E.; Llanos, J.; Sáez, D. Energy management system for a renewable based microgrid with a demand side management mechanism. In proceedings of the IEEE Symposium on Computational Intelligence Applications in Smart Grid 2011, Paris, France, 11–15 April 2011; pp. 1–8. https://doi.org/10.1109/CIASG.2011.5953338.
9. Rodríguez, M.; Salazar, A.; Arcos-Aviles, D.; Llanos, J.; Martínez, W.; Motoasca, E. A Brief Approach of Microgrids Implementation in Ecuador: A Review. In *Lecture Notes in Electrical Engineering*; Springer: Cham, Switzerland, 2021; Volume 762, pp. 149–163. https://doi.org/10.1007/978-3-030-72208-1_12.
10. Llanos, J.; Morales, R.; Núñez, A.; Sáez, D.; Lacalle, M.; Marín, L.G.; Hernández, R.; Lanas, F. Load estimation for microgrid planning based on a self-organizing map methodology. *Appl. Soft Comput.* **2017**, *53*, 323–335. https://doi.org/10.1016/j.asoc.2016.12.054.
11. Caquilpan, V.; Saez, D.; Hernandez, R.; Llanos, J.; Roje, T.; Nunez, A. Load estimation based on self-organizing maps and Bayesian networks for microgrids design in rural zones. In Proceedings of the 2017 IEEE PES Innovative Smart Grid Technologies Conference - Latin America (ISGT Latin America), Quito, Ecuador, 20–22 September 2017, pp. 1–6. https://doi.org/10.1109/ISGT-LA.2017.8126709.
12. Palma-Behnke, R.; Benavides, C.; Lanas, F.; Severino, B.; Reyes, L.; Llanos, J.; Saez, D. A microgrid energy management system based on the rolling horizon strategy. *IEEE Trans. Smart Grid* **2013**, *4*, 996–1006. https://doi.org/10.1109/TSG.2012.2231440.
13. Rey, J.M.; Vera, G.A.; Acevedo-Rueda, P.; Solano, J.; Mantilla, M.A.; Llanos, J.; Sáez, D. A Review of Microgrids in Latin America: Laboratories and Test Systems. *IEEE Lat. Am. Trans.* **2022**, *20*, 1000–1011. https://doi.org/10.1109/TLA.2022.9757743.
14. Javier, G.; Quevedo-Nolasco, A.; Castro-Popoca, M.; Arteaga-Ramírez, R.; Vázquez-Peña, M.A.; Zamora-Morales, B.P.; Aguado-Rodríguez, G.J.; Quevedo-Nolasco, A.; Castro-Popoca, M.; Arteaga-Ramírez, R.; et al. PREDICCIÓN DE VARIABLES METEOROLÓGICAS POR MEDIO DE MODELOS ARIMA. *Agrociencia* **2016**, *50*, 1–13.
15. Viraj A. Gulhane, S.V.R.& C.B.P. Correlation Analysis of Soil Nutrients and Prediction Model Through ISO Cluster Unsupervised Classification with Multispectral Data. *Springer Link* **2022**, *82*, 2165–2184. https://doi.org/10.1007/s11042-022-13276-2.
16. Pande, C.B.; Al-Ansari, N.; Kushwaha, N.L.; Srivastava, A.; Noor, R.; Kumar, M.; Moharir, K.N.; Elbeltagi, A. Forecasting of SPI and Meteorological Drought Based on the Artificial Neural Network and M5P Model Tree. *Land* **2022**, *11*, 2040. https://doi.org/10.3390/land11112040
17. Mora Cunllo, V.E. Diseño e implementación de un modelo software basado en técnicas de inteligencia artificial, para predecir el índice de radiación solar en Riobamba-Ecuador. 2015. Available online: http://repositorio.espe.edu.ec/bitstream/21000/12216/1/T-ESPEL-MAS-0027.pdf (accessed on 24 November 2022).
18. Universitario, S.; Estad, E.N.; Aplicada, S.; Fern, R.A.; Javier, F.; Morales, A. *Series Temporales Avanzadas: Aplicación de Redes Neuronales para el Pronóstico de Series de Tiempo*; Universidad de Granada: Granada, Spain, 2021.
19. Raschka, S.; Patterson, J.; Nolet, C. Machine learning in python: Main developments and technology trends in data science, machine learning, and artificial intelligence. *Information* **2020**, *11*, 193. https://doi.org/10.3390/info11040193.
20. Carlos, J.; Rodriguez, M. Desarrollo de una herramienta inteligente centrada en visión plantaciones de arroz, usando lenguaje de programación Python. Ph.D. Thesis, Universidad de Guayaquil: Guayaquil, Ecuador, 2022.
21. Ben Bouallègue, Z.; Cooper, F.; Chantry, M.; Düben, P.; Bechtold, P.; Sandu, I. Statistical modelling of 2m temperature and 10m wind speed forecast errors. *Mon. Weather. Rev.* **2022**. https://doi.org/10.1175/MWR-D-22-0107.1 Available online: https://journals.ametsoc.org/view/journals/mwre/aop/MWR-D-22-0107.1/MWR-D-22-0107.1.xml (accessed on 18 January 2023).
22. Montero Granados, R. Modelos de Regresión Lineal Múltiple. Technical Report; Documentos de Trabajo en Economía Aplicada; Universidad de Granada: Granada, Spain, 2006.
23. Aurélien, G. *Hands-on Machine Learning with Scikit-Learn & Tensorflow*; O'Reilly Media, Inc.: Sebastopol, CA, USA, 2017.

24. Elbeltagi, A.; Kumar, M.; Kushwaha, N.L.; Pande, C.B.; Ditthakit, P.; Vishwakarma, D.K.; Subeesh, A. Drought indicator analysis and forecasting using data driven models: Case study in Jaisalmer, India. *Stoch. Environ. Res. Risk Assess.* **2022**, *37*, 113–131. https://doi.org/10.1007/s00477-022-02277-0.
25. B, M.L.; Topolski, B.; Mazurek, M. Application of XGBoost Algorithm. *Data Anal.* **2017**, *10244*, 661–671. https://doi.org/10.1007/978-3-319-59105-6.
26. Menacho Chiok, C.H. Modelos de regresión lineal con redes neuronales. *An. Científicos* **2014**, *75*, 253. https://doi.org/10.21704/ac.v75i2.961.
27. Popescu, M.C.; Balas, V.E.; Perescu-Popescu, L.; Mastorakis, N. Multilayer perceptron and neural networks. *WSEAS Trans. Circuits Syst.* **2009**, *8*, 579–588.
28. Soto-Bravo, F.; González-Lutz, M.I. Analysis of statistical methods to evaluate the performance of simulation models in horticultural crops. *Agron. Mesoam.* **2019**, *30*, 517–534. https://doi.org/10.15517/am.v30i2.33839.
29. Gopi, A.; Sharma, P.; Sudhakar, K.; Ngui, W.K.; Kirpichnikova, I.; Cuce, E. Weather Impact on Solar Farm Performance: A Comparative Analysis of Machine Learning Techniques. *Sustainability* **2023**, *15*, 439. https://doi.org/10.3390/su15010439.
30. de Myttenaere, A.; Golden, B.; Le Grand, B.; Rossi, F. Mean Absolute Percentage Error for regression models. *Neurocomputing* **2016**, *192*, 38–48. https://doi.org/10.1016/j.neucom.2015.12.114.
31. Chai, T.; Draxler, R.R. Root mean square error (RMSE) or mean absolute error (MAE)? -Arguments against avoiding RMSE in the literature. *Geosci. Model Dev.* **2014**, *7*, 1247–1250. https://doi.org/10.5194/gmd-7-1247-2014.