



Research article

Examination of empirical and Machine Learning methods for regression of missing or invalid solar radiation data using routine meteorological data as predictors

Konstantinos X Soulis*, Evangelos E Nikitakis, Aikaterini N Katsogiannou and Dionissios P Kalivas

GIS Research Unit, Laboratory of Soil Science and Agricultural Chemistry, Department of Natural Resources Management and Agricultural Engineering, Agricultural University of Athens, Iera Odos 75, Athens, 11855, Greece

* **Correspondence:** Email: soco@aua.gr; Tel: +302105294070.

Abstract: Sensors are prone to malfunction, leading to blank or erroneous measurements that cannot be ignored in most practical applications. Therefore, data users are always looking for efficient methods to substitute missing values with accurate estimations. Traditionally, empirical methods have been used for this purpose, but with the increasing accessibility and effectiveness of Machine Learning (ML) methods, it is plausible that the former will be replaced by the latter. In this study, we aimed to provide some insights on the state of this question using the network of meteorological stations installed and operated by the GIS Research Unit of the Agricultural University of Athens in Nemea, Greece as a test site for the estimation of daily average solar radiation. Routine weather parameters from ten stations in a period spanning 1,548 days were collected, curated, and used for the training, calibration, and validation of different iterations of two empirical equations and three iterations each of Random Forest (RF) and Recurrent Neural Networks (RNN). The results indicated that while ML methods, and especially RNNs, are in general more accurate than their empirical counterparts, the investment in technical knowledge, time, and processing capacity they require for their implementation cannot constitute them as a panacea, as such selection for the best method is case-sensitive. Future research directions could include the examination of more location-specific models or the integration of readily available spatiotemporal indicators to increase model generalization.

Keywords: machine learning; solar radiation; recurrent neural networks; random forest; empirical methods; regression

Abbreviations: ML: Machine Learning; RF: Random Forest; RNN: Recurrent Neural Networks; GIS: Geographic Information Systems; LSTM: Long-Short Term Memory

1. Introduction

Solar radiation is the energy source of the planet, and knowledge of its magnitude on any given spatiotemporal frame is necessary for a multitude of applications, from power generation [1] to irrigation planning [2]. Focusing on its agricultural applications, and with the advent of smart agriculture, weather stations connected to the Internet are often used to supply these important measurements directly to a monitoring team tens or hundreds of kilometers away. However, an uncommon but persistent issue of operating a meteorological station at a remote location is the loss of data pertaining to malfunctioning sensors. Due to the difficulty of coordination and access, it could take days for a maintenance team to fix the problem after it is first detected, wasting precious data and poking holes in otherwise complete datasets. This issue requires an accurate yet easy to implement solution, as complex methodologies may be unappealing or wasteful for tackling such a mundane issue.

A similar problem is the absence of solar radiation measurements in general meteorological observation stations observed in studies like those of Zang et al. [3,4], Ağbulut et al. [5], and Soulis et al. [6]. The absence of solar radiation measurements is even more pronounced in the case of historical data [6]. In all these cases, estimation of the solar radiation values through the more readily available data for other meteorological parameters such as air temperature and air relative humidity was necessary in some capacity, either through empirical functions as proposed by Hargreaves and Samani [7] or Meza and Yebra [8], or, more recently, through Machine Learning (ML) algorithms [3,5].

The studies presented in Table 1 offer a review of previous studies presenting robust methodologies for predicting daily average global solar radiation based on other meteorological variables. These studies incorporate ML methods, either as individual models [9–11] or hybrid ML-empirical models [4], reporting promising results. Some researchers [5,12,13] have even performed comparisons of different ML methods to discern the best one to use for tackling this problem. However, literature pertaining to comparing the performance of traditional empirical methods and the relatively new ML methods, taking into account the availability of the required variables, the methods' simplicity, and the computational requirements in this setting, seems scarce. Such an evaluation is a worthwhile endeavor since ML methods are inherently less accessible compared to the well-established and intensely calibrated traditional methods, with many researchers opting to use the latter over the former for the sake of convenience.

Table 1. Selection of studies using Machine Learning methods for the estimation of solar radiation.

Publication	Machine Learning Methods Used
[4]	Artificial Neural Networks
[5]	Support Vector Machines, Artificial Neural Networks, Kernel Nearest Neighbor
[9]	Genetic Programming
[10]	Random Forest
[11]	Random Forest
[12]	Support Vector Machines, Artificial Neural Networks, Adaptive Network-based Fuzzy Inference System, Multiple Linear Regression
[13]	Support Vector Machines, Artificial Neural Networks, Kernel Nearest Neighbor, Gaussian Process Regression, Extreme Learning Machines

We aim to compare the accuracy of different methods for the estimation of daily global solar radiation using other routine meteorological variables for the purposes of filling in missing measurements while considering the differences in availability of measured parameters for any given station, simplicity, and computational requirements. Considering the focus on simplicity and speed of this analysis, the methods selected for this comparison were the well-established empirical equation proposed by Hargreaves and Samani [7] and the modified version of the same equation proposed by Valiantzas [14]. These traditional methods were compared with various implementations of two ML models. These models were the Random Forest, which was selected for its relatively simpler implementation without sacrificing accuracy [15], and the Recurrent Neural Networks, which was selected for its specialization in handling sequential data [16].

2. Materials and methods

2.1. Study area, data sourcing and processing

The data used in this paper came from the meteorological station network operated by the GIS Research Unit of the Agricultural University of Athens in the area of Nemea, comprising ten (10) stations (Figure 1) that record data on precipitation, temperature, wind speed, relative humidity, and solar radiation on a 15-minute time step, which were drawn from the period from 1/10/2019 to 27/12/2023.

This data was then used to calculate daily sums for precipitation and daily average, maximum, and minimum values for temperature, relative humidity, wind speed, and solar radiation. The aforementioned process was completed using a semi-automatic algorithm developed in MS Excel VBA to reduce processing time. Time steps that had missing or evidently erroneous measurement values based on simple criteria considering the acceptable value range for each parameter were entirely omitted from the procedure.

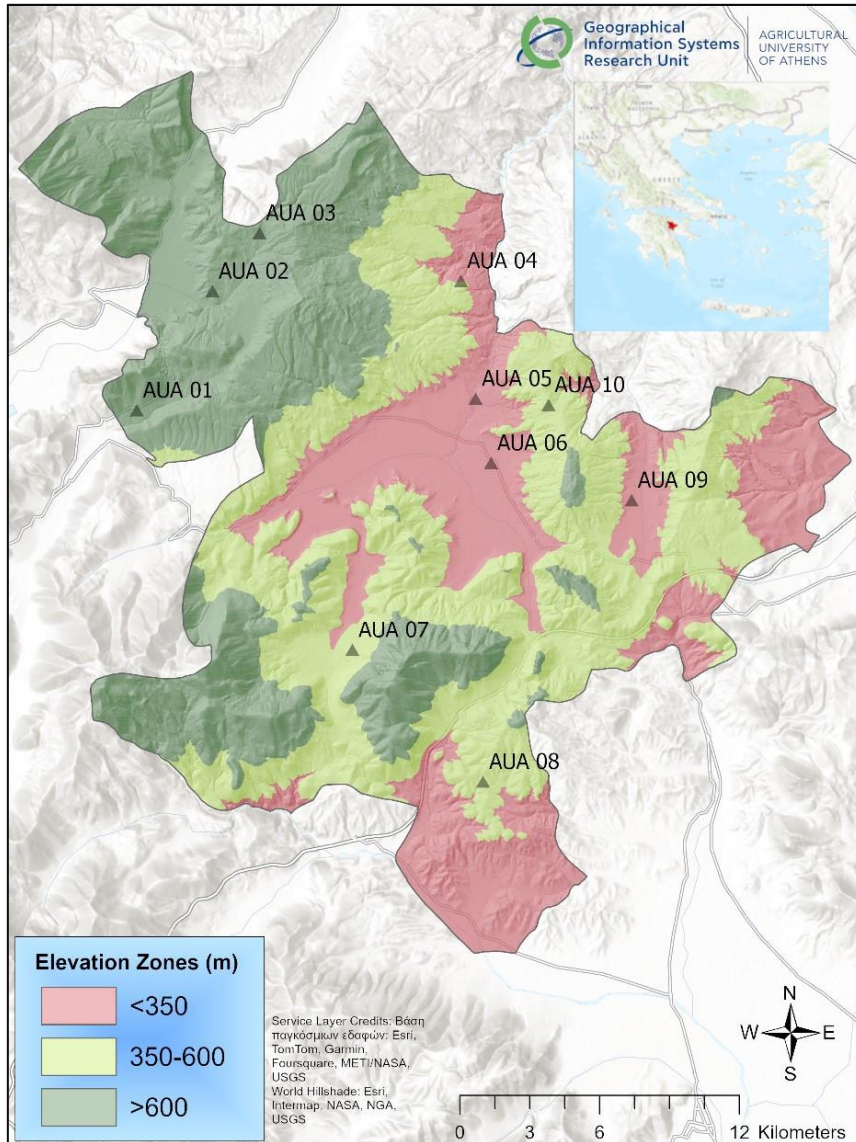


Figure 1. The study area (Nemea) and the locations of the meteorological stations within it.

Daily extraterrestrial radiation is a very important variable in estimating average daily solar radiation [7,14], and its values were calculated in MS Excel VBA for each of the stations’ locations for all selected dates using the following equation [2]:

$$R_a = \frac{24(60)}{\pi} G_{sc} d_r [\omega_s \sin(\varphi) \sin(\delta) + \cos(\varphi) \cos(\delta) \sin(\omega_s)] \quad (1)$$

where R_a is the extraterrestrial radiation in $\text{MJ m}^{-2} \text{ day}$, G_{sc} is the solar constant $0.0820 \text{ MJ m}^{-2} \text{ day}$, d_r is the inverse relative distance Earth-Sun, ω_s is the sunset hour angle in radians, φ is the latitude in radians, and δ is the solar declination angle in radians.

All the variables were then investigated for correlation with average solar radiation using IBM SPSS Statistics 25. The correlation coefficients used were Pearson’s and Spearman’s ranks to capture both linear and non-linear relationships, and the results are presented in Table 2. The variables that showed a significantly high correlation with average solar radiation with both coefficients across all

ten locations were extraterrestrial radiation, average relative humidity, maximum, minimum, and average temperature; average wind speed showed a high significance level for some stations, but gave significance values over but very close to 0.05 for others and it was decided that it would not be omitted. Furthermore, the same parameters are used in existing empirical methods [7,14]. Consequently, these were the characteristics that could potentially be used for the interpolation of average solar radiation by the ML algorithms used in this paper.

Table 2. Correlation coefficients and significance for daily average solar radiation.

Variable		Pearson's	Spearman's
Maximum Temperature	Coefficient	0.7420	0.7705
	Significance	≈0	≈0
Minimum Temperature	Coefficient	0.5556	0.5627
	Significance	≈0	≈0
Average Temperature	Coefficient	0.7184	0.7165
	Significance	≈0	≈0
Average Wind Speed	Coefficient	0.1885	0.3234
	Significance	0.0465	0.0586
Average Relative Humidity	Coefficient	-0.7926	-0.7960
	Significance	≈0	≈0
Extraterrestrial Solar Radiation	Coefficient	0.8306	0.8245
	Significance	≈0	≈0

Another piece of data that was needed and had to be calculated was the dew point temperature, which was calculated in daily temporal step in MS Excel using the formula by Allen et. al (1998) [2]:

$$T_{dew} = \frac{116.91 + 237.3 \ln(e_a)}{16.78 - \ln(e_a)} \quad (2)$$

where e_a is the actual vapor pressure in kPa and was calculated by the following formula:

$$e_a = 0.6108 \times \exp\left(\frac{17.27 T}{T + 237.3}\right) \times \frac{RH}{100} \quad (3)$$

where RH is the relative humidity as %, T is the average daily temperature in °C.

The data was then split in two datasets, the data from 1/10/2019 to 30/9/2022 was used for the training and calibration of the machine learning models and relevant empirical equations versions, while the data from 1/10/2022 to 27/12/2023 was used for the calibration of the relevant equation versions and for validation and comparison of both the equations and machine learning models. Given ten locations, the training dataset comprised of a total of 10,970 data points and the validation and comparison dataset was comprised of 4,499 data points.

2.2. Empirical equations methods

Some of the equations traditionally used in the estimation of missing solar radiation values and which will be examined in this paper are the equation proposed by Hargreaves and Samani (Eq. 4) [7] and the modified version of the same equation proposed by Valiantzas (Eq. 5) [14], which uses dew point temperature in place of minimum temperature.

$$R_s = k_{RS} \times R_a (T_{max} - T_{min})^{0.5} \quad (4)$$

$$R_s = k_{RS} \times R_a (T_{max} - T_{dew})^{0.5} \quad (5)$$

where R_s is the solar radiation in Wm^{-2} , R_a is the extraterrestrial radiation in Wm^{-2} , T_{max} is the maximum daily temperature in $^{\circ}\text{C}$, T_{min} is the minimum daily temperature in $^{\circ}\text{C}$, T_{dew} is the calculated dew point temperature in $^{\circ}\text{C}$, and k_{RS} is an empirical solar radiation adjustment coefficient that generally varies from 0.12 to 0.25.

Both equations were examined and split in three use cases each. The first use case dubbed “non-calibrated” was adopting the conventional k_{RS} value of 0.17 for all stations, the second use case dubbed “local” was calibration of the k_{RS} values [17] for each of the ten locations separately using the data from the training dataset and the third dubbed “total” was calibrated simultaneously using the data of all ten locations. The calibration was done up to the third decimal digit of k_{RS} and was performed by slightly adjusting the value until the corresponding equation’s result gave the lowest Root Mean Square Error (RMSE) value when compared to the ground truth data of the training dataset. RMSE is calculated as follows [4]:

$$RMSE = \sqrt{\frac{\sum_{i=1}^n (y_i - y_{pi})^2}{n}} \quad (6)$$

where y_i is the observed value of the i -th time step, y_{pi} is the predicted value of the i -th timestep, and n is the number of time steps.

2.3. Random forest

Out of many possible options, the Random Forest (RF) algorithm was chosen for its robustness in dealing with regression problems and its relative ease of development and use. Using RStudio, three RF models were constructed, each using a diminishing number of input variables. The first iteration, dubbed “RFcomplete”, used all the variables found to have a significant correlation with average solar radiation, as explained in section 2.1, namely maximum temperature, minimum temperature, average temperature, average wind speed, average relative humidity, and extraterrestrial solar radiation. The second model, dubbed “RFhalf”, did away with average wind speed and temperature to be comparable to Equation 5 in terms of prerequisite data. The final, dubbed “RFminimal”, would use only extraterrestrial solar radiation and maximum and minimum temperature to be in direct juxtaposition to Equation 4 and its minimal need for data.

Using the grid search method, different models with different combinations of hyperparameters were trained for each iteration in order to discern the best performing combination in terms of R^2 on

the independent validation dataset. The coefficient of determination (R^2) is defined as the variability explained by the regression model and is calculated by the following formula [18]:

$$R^2 = 1 - \frac{\sum_{i=1}^n (y_i - y_{pi})^2}{\sum_{i=1}^n (y_i - \bar{y})^2} \quad (7)$$

where y_i is the observed value of the i -th time step, y_{pi} is the predicted value of the i -th timestep, and \bar{y} is the average of all observed values.

The hyperparameters in question were the number of trees, the minimum terminal node size, the sample size for each tree, the depth of the decision trees, and the number of variables available for splitting at each tree node. The hyperparameters chosen for each model are presented in Table 3.

Table 3. Final Random Forest model hyperparameters.

Model	Number of Trees	Minimum Node Size	Sample Size	Maximum Tree Depth	Variable Number for Splitting
RFcomplete	400	2	0.6	10	4
RFhalf	400	4	0.6	10	3
RFminimal	600	2	0.2	10	3
Available options	200, 400, 600, 800, 1000	2, 4, 6	0.2, 0.4, 0.6, 0.8	5, 10, 15, 30	2, 3, 4, 5, 6

The first two models favored less random trees but less dense forests, while the RFminimal model was most effective with a denser but more random forest. This could potentially be explained by the number and importance of the variables fed to each model, as shown in Figure 2. With a higher number of more important input variables, the average tree in the former models is more robust as a predictor than their RFminimal counterparts [19]; thus, fewer trees are needed to get an accurate prediction, and a median degree of randomness is required as to not compromise that robustness while allowing for variability [20,21]. Inversely, a more randomized forest with higher density makes sense if each single decision tree is relatively weaker. This interpretation is supported by comparing the importance of each feature for its respective model, as average relative humidity was identified as the second most important feature in both models, where it is used as a predictor.

Additionally, all of the models were most effective when choosing a number of trees in the middle-lower part of the range of available options (200 to 1000 trees). This is an extension of the previous observation, suggesting that given a high number of trees with medium or low randomness, the model overfits the training dataset without adding substantially to the accuracy [22], while highly random trees are either too inefficient as learners or require a forest density beyond the range allowed in our training to reach comparable accuracy levels [23].

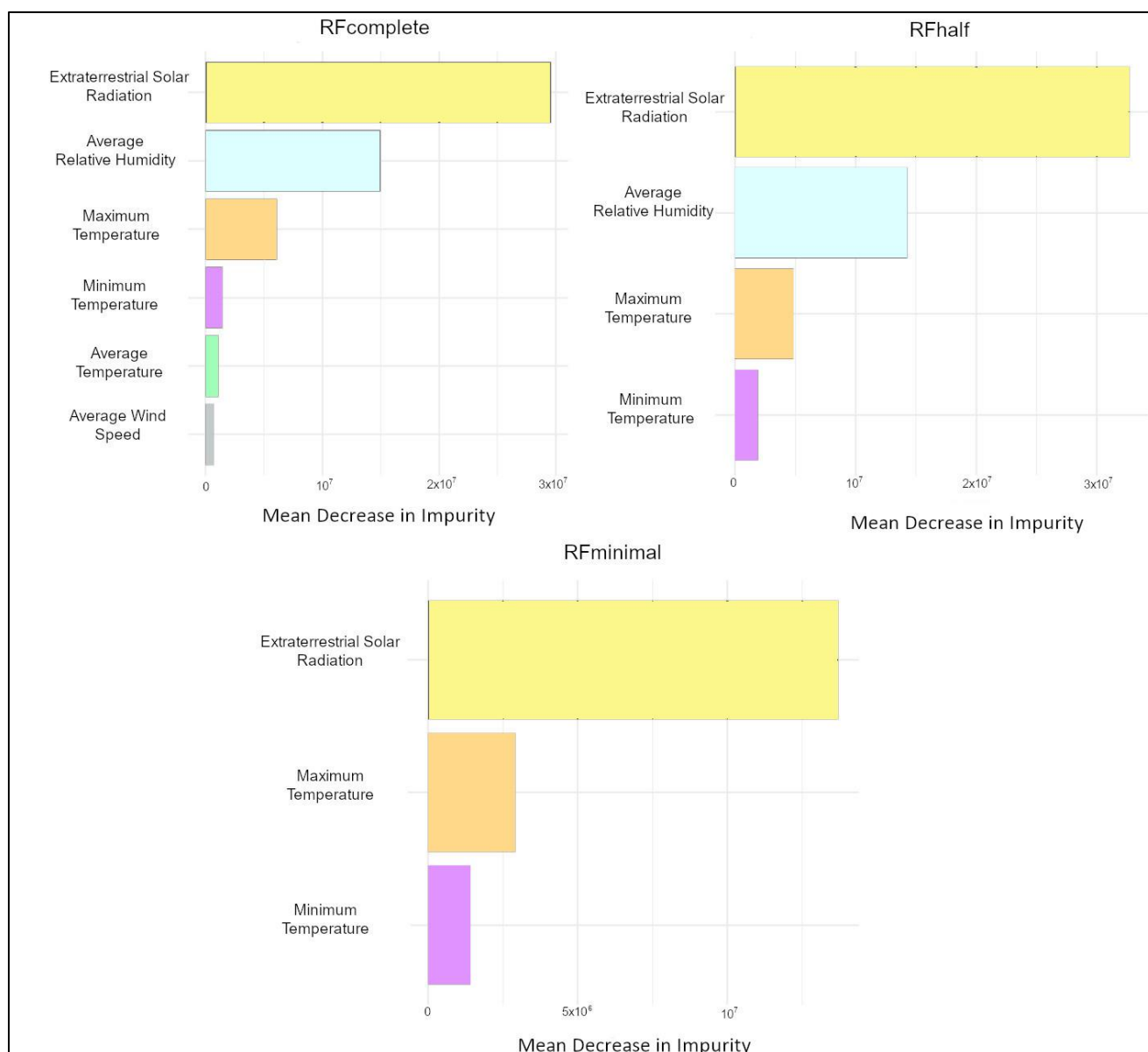


Figure 2. Mean Decrease in Impurity graphs for the visualization of the importance of each feature in each respective Random Forest model iteration.

2.4. Recurrent neural networks

Artificial Neural Networks are computational models that simulate the workings of the human brain to make decisions [16] and are often portrayed as one of the most accurate machine learning methods available [24]. There are various ANN architectures that specialize in different applications, specifically for regression problems dealing in temporal sequences, and Recurrent Neural Networks (RNNs) appear to be the most fitting choice [25]. This RNN used a Long Short Term Memory (LSTM) layer as the recipient of the input, followed by a dropout layer and a number of hidden layers, comprised of one dense layer followed by a dropout layer each. To avoid features with more massive numerical values having an inordinate effect on the learning process, Z-score normalization

is performed on each feature of the data [26] before it is fed into the LSTM layer using the following equation:

$$Z_i = \frac{X_i - \bar{X}}{S} \quad (8)$$

where Z_i is the normalized value of the i -th observation, X_i is the original value of the i -th observation, \bar{X} is the mean value of all observations, and S is the standard deviation of all observations.

Three RNN iterations were developed using RStudio, dubbed “ANNcomplete”, “ANNhalf”, and “ANNminimal”, categorized with respect to the features used in accordance with their respective RF counterparts, as explained in the previous section. Similarly, different models with different combinations of hyperparameters were trained for each respective iteration using grid search, selecting the best-performing combination in terms of R^2 on the independent validation dataset. These hyperparameters were the number of neurons for each layer, the number of hidden layers, the dropout rate, and the number of epochs as, shown in Table 4.

Table 4. Final Recurrent Neural Network model hyperparameters.

Model	Number of Layers and Neurons of Each Layer	Dropout Rate	Epochs
ANNcomplete	LSTM: 64 2 Hidden: 32 and 64	0.5	200
ANNhalf	LSTM: 32 2 Hidden: 32 and 64	0.2	200
ANNminimal	LSTM: 64 2 Hidden: 16 and 32	0.2	200
Available options	LSTM: 32, 64, 128 2 or 3 Hidden: (32, 64), (16, 32), (64, 128), (16, 32, 64)	0.2, 0.5, 0.8	200, 350, 500

Foregoing the inclusion of time steps in the hyperparametrization process appears like an odd choice for the training of RNNs. However, to allow for maximum flexibility in the application of the model to temporal datasets of wildly different lengths, the time step hyperparameter was set to 1 [27], as the LSTM layer should be capable of capturing long-term temporal dependencies on its own [28]. Additionally, it is important to mention that the layer architecture for each RNN model is identical, as shown in Figure 3.

The final hyperparameters seem to suggest a relatively simple relationship between the selected features and solar radiation, as all models performed best on the lowest available number of epochs and hidden layers. Furthermore, the “ANNcomplete” model favored a higher dropout rate than its counterparts using a smaller number of predictor features, possibly because the additional features, which were explained to be less significant in the preceding section, led to misestimating connection weights, and thus a higher deactivation rate allows for consistent use of the more robust nodal connections [29]. However, it could also be explained by the comparatively lower number of total neurons in the other two models, though marginally, as it could afford to deactivate more neurons [30].

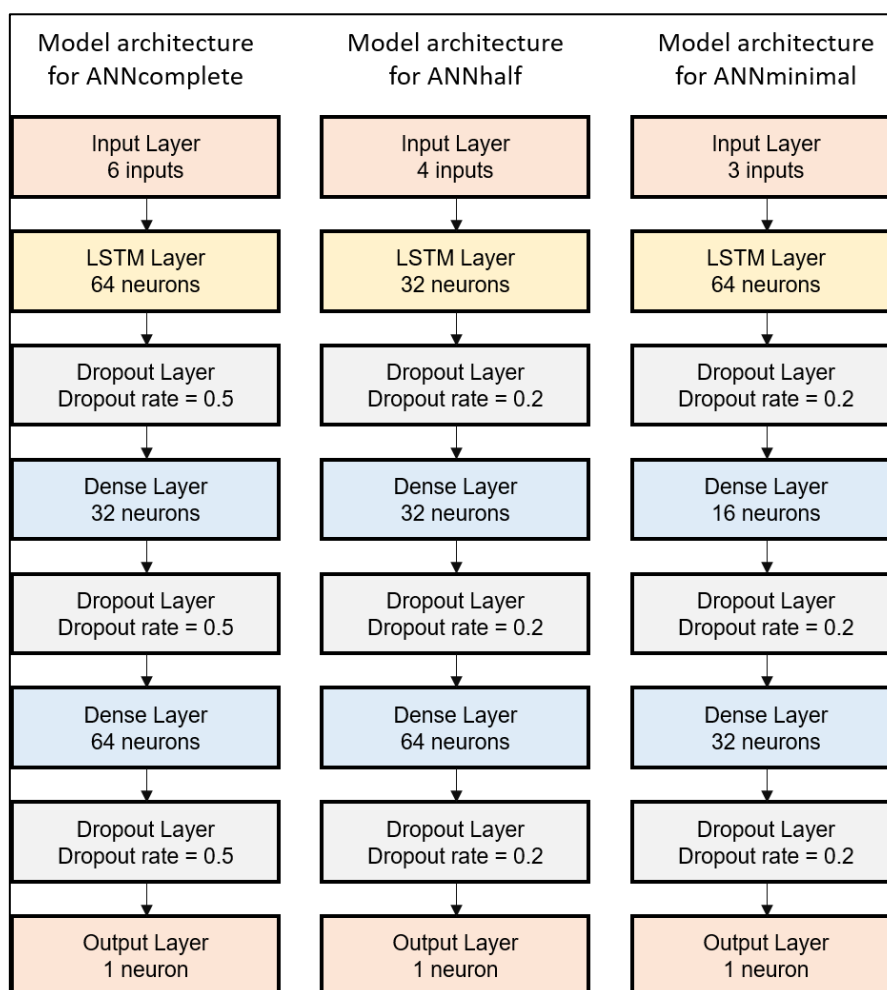


Figure 3. Recurrent Neural Network models architecture.

Speaking of neurons, no model achieved better accuracy with the highest possible number of neurons, which would suggest that noise within the dataset is quite prevalent [30]. Furthermore, as expected, the model with the lowest number of predictor variables also has the lowest available number of neurons for its hidden layers, though the highest number was picked for the LSTM layer. The “ANNhalf” model has a lower number of neurons in the LSTM layer, and given that relative humidity is a major predictor in this model but missing in the “ANNmin” model, it would be fair to assume that the relationship between relative humidity and average daily solar radiation isn’t particularly dependent on long-term interactions [28].

2.5. Metrics for comparison

Different metrics were used to judge the performance of the final models from different perspectives [4], as, for example, RMSE penalizes errors of greater magnitude more severely than Mean Absolute Error (MAE), which could be misleading [31] but could also potentially reveal model differences more clearly [32]. In addition to R^2 and RMSE, which were discussed above and calculated using Eq.7 and Eq.6, respectively, additional metrics were used to compare the results. These are as follows:

Relative Root Mean Square (rRMSE) which was calculated as [4,33]:

$$rRMSE = 100 \frac{\sqrt{\frac{\sum_{i=1}^n (y_i - y_{pi})^2}{n}}}{\bar{y}} \quad (9)$$

Mean Absolute Error (MAE) which was calculated as [32]:

$$MAE = \frac{\sum_{i=1}^n |y_i - y_{pi}|}{n} \quad (10)$$

Mean Bias Error (MBE) which was calculated as [31]:

$$MBE = \frac{\sum_{i=1}^n (y_i - y_{pi})}{\bar{y}} \quad (11)$$

where y_i is the observed value of the i -th time step, y_{pi} is the predicted value of the i -th timestep, n is the number of time steps, and \bar{y} is the average of all observed values.

For RMSE, rRMSE and MAE, values closer to 0 denote a better performance, for R^2 this is reversed with values closer to 1 denoting higher accuracy. MBE values are indicative of positive or negative bias, with values closer to 0 denoting a more equal distribution of overestimations and underestimations and not necessarily higher model accuracy.

3. Results and Discussion

On the totality of the validation dataset, the ANNcomplete model (RMSE = 41.25, $R^2 = 0.810$) showed the best results in all metrics, followed by the ANNhalf model (RMSE = 43.52, $R^2 = 0.789$). The ANNminimal model (RMSE = 48.20, $R^2 = 0.740$) had a marginally worse performance than the RFcomplete model (RMSE = 47.39, $R^2 = 0.746$) and performed almost equally to the RFhalf model (RMSE = 47.93, $R^2 = 0.740$). The non-calibrated equation using T_{dew} [14] (RMSE = 48.75, $R^2 = 0.736$) performed similarly to the aforementioned, scoring marginally worse in all metrics except for MAE, where it had a slight edge. The next cluster of comparable performers (in order of descending accuracy) was comprised of the calibrated by location T_{dew} equation (RMSE = 50.85, $R^2 = 0.713$), the calibrated by location T_{min} equation (RMSE = 51.10, $R^2 = 0.710$), and the RFminimal model (RMSE = 51.71, $R^2 = 0.697$). Finally, the worst performers were the non-calibrated T_{min} equation (RMSE = 53.18, $R^2 = 0.686$), and the T_{dew} (RMSE = 53.00, $R^2 = 0.687$) and T_{min} (RMSE = 54.12, $R^2 = 0.674$) equations calibrated on the totality of locations. All the corresponding results are presented in Table 5.

As for bias, all three ANN models were geared towards minor underestimation, with the ANNcomplete iteration having an almost equal distribution of over and underestimations. The Random Forest models, on the other hand, consistently underestimated average solar radiation, with all but the RFminimal model holding the lowest negative MBE scores (Table 5). The calibrated variations of the equations using T_{min} were almost equally as prone to underestimation as the RFcomplete and RFhalf models, with the respective T_{dew} equations also gearing towards underestimation but of a smaller magnitude. Finally, both non-calibrated equations were the only methods to have positive MBE scores, though the magnitude of overestimation was on par with the middle-higher part of the spectrum of underestimation values.

Table 5. Comparison of metrics for all models on the totality of the validation dataset. The scores were obtained from the combination of all 4,499 time steps of the validation dataset.

	RMSE	R ²	rRMSE	MAE	MBE
Local Equation T _{min}	51.10	0.710	28.58%	40.67	-15.17
Local Equation T _{dew}	50.85	0.713	28.45%	41.37	-22.06
Total Equation T _{min}	54.12	0.674	30.30%	42.02	-14.62
Total Equation T _{dew}	53.00	0.687	29.67%	42.14	-22.28
Non-calibrated Equation T _{min}	53.18	0.686	29.75%	40.44	15.89
Non-calibrated Equation T _{dew}	48.75	0.736	27.26%	36.51	14.94
RFcomplete	47.39	0.746	26.42%	37.96	-22.64
RFhalf	47.86	0.740	26.73%	38.39	-22.88
RFminimal	51.71	0.697	28.84%	41.30	-17.84
ANNcomplete	41.25	0.810	22.84%	31.05	-1.19
ANNhalf	43.52	0.789	24.10%	33.55	-6.28
ANNminimal	47.86	0.740	26.73%	37.03	-7.04
Legend		Best Performance		Worst Performance	
Legend	High Negative Bias				High Positive Bias

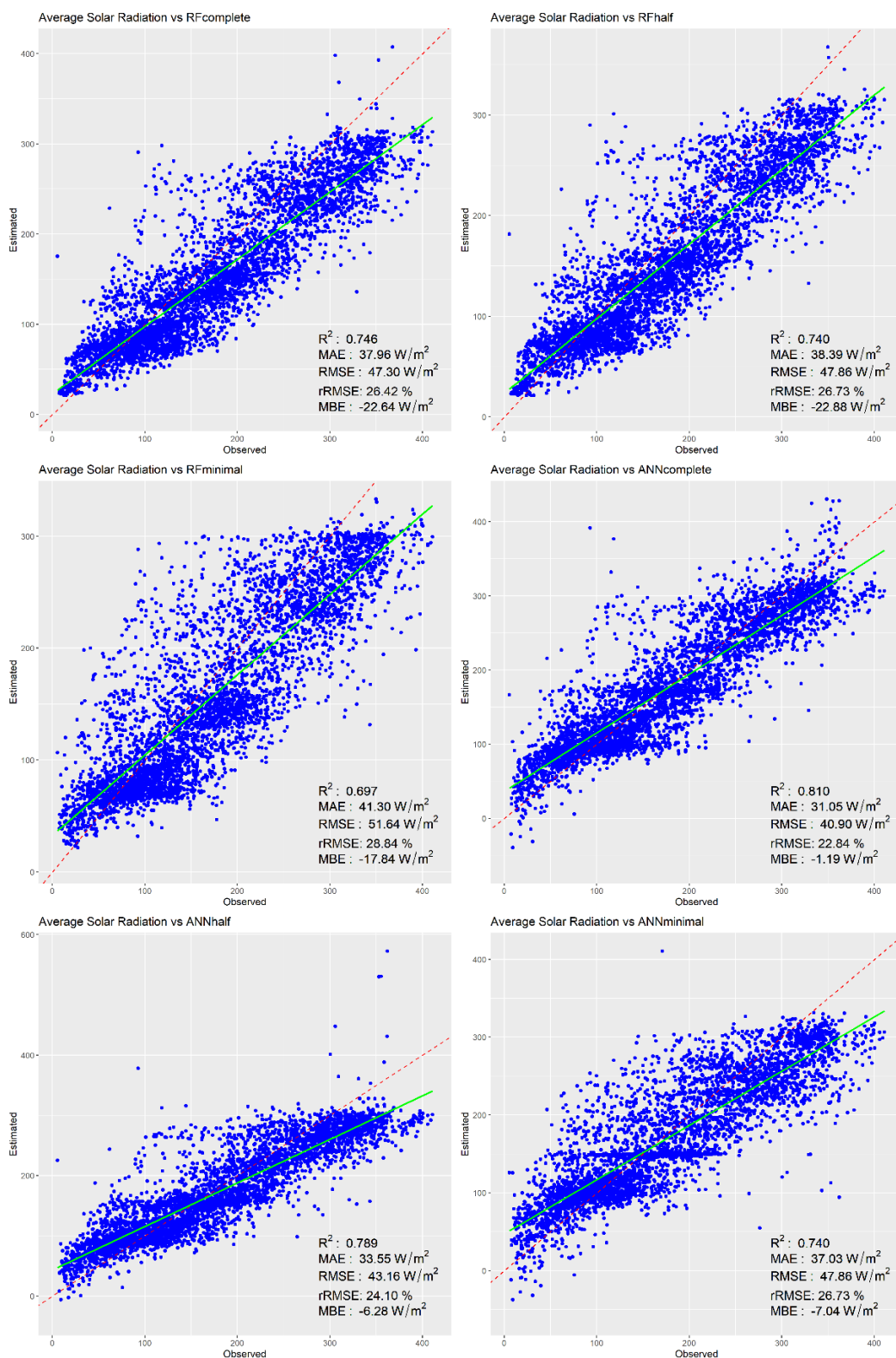


Figure 4. Observed vs estimated values. Scatter plots for the Machine Learning models on the totality of the validation dataset. The red line represents a perfect prediction, where at every point on the line, Estimated y equals Observed x, and the green line is the scatter trendline.

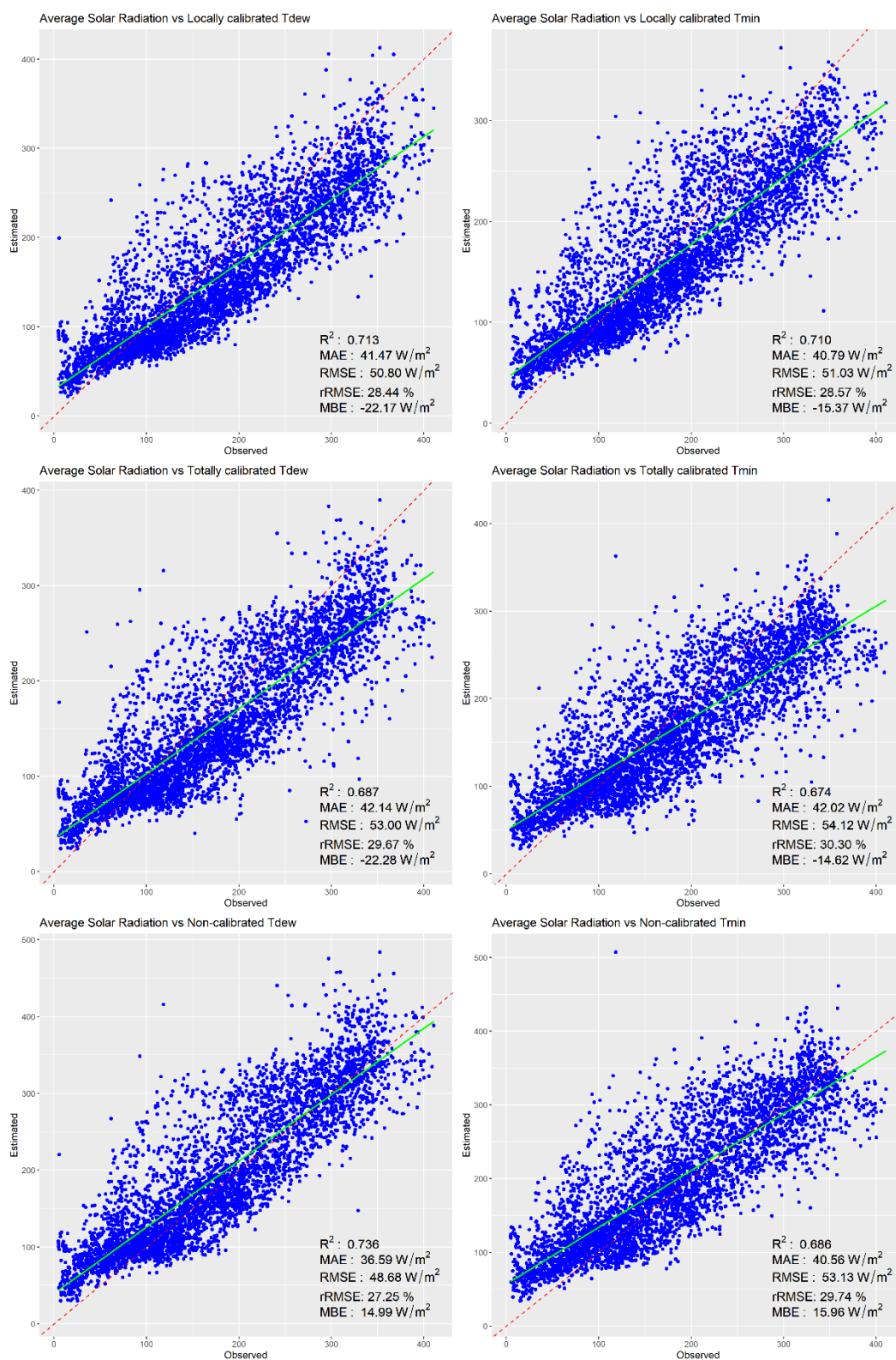


Figure 5. Observed vs estimated values scatter plots for the iterations of Eq.3 and Eq.4 for the totality of the validation dataset. The red line represents a perfect prediction, where at every point on the line, Estimated y equals Observed x, and the green line is the scatter trendline.

Taking a deeper look at the results (Table 5), it is apparent that the RNN models consistently outperformed the RF models using the same or even more predictor variables, with the expected “weakest” RNN model having an almost equal performance to the middle RF model in all metrics but MAE where the RNN performed better, which could be explained by the inherent limitations of RF algorithms versus RNNs in handling regression tasks. Due to the inner workings of RF models, predictions tend to form clusters in a stepwise fashion [34], essentially creating biases against predicting values between these clusters, which could contribute to a significant loss of accuracy on the total dataset. This behavior is evident in Figure 4, where, in the respective plots, areas of lower point density can be observed between areas of higher point density more frequently than in their RNN counterparts. For comparison, the respective scatterplots for the equation methods (Figure 5) show continuous scatter clouds with no evident stepwise clustering.

Furthermore, as Random Forest relies on averaging the predictions of multiple decision trees, the final predictions tend to be stabilized closer to the expected values reducing variability [19], which on one hand reduces the risk of overfitting, but on the other, it creates a relative weakness in predicting extreme or rare values [35]. Figure 4 again seems to support this idea, as the plots for the RF models’ scatters seem to present a steeper incline with the perfect prediction line towards the higher end of values compared to their RNN counterparts. That is not to say that RNNs are not subject to the aforementioned observations, as both a degree of stepwise changes of prediction and a drop off of accuracy at the extremes are visible. In fact, the latter is much more pronounced on the lower end of values on RNNs compared to the RFs, with a handful of predictions even being negative values, which is a potential sign that some of the weights leading to the reduction of the value of the final prediction were overestimated leading to unrealistic results. Given that the number of negative values was negligible compared to the total sample size, this does not appear to be a cause for major concern, especially since they seem to be restricted to dates and locations that measured surprisingly low daily average solar radiation values. However, this should be further studied in future research.

However, the discrepancy in accuracy could be explained better, not by the RF’s shortcomings, as they seem to be shared between the two techniques to some degree, but by the RNNs’ ability to recognize long-term dependencies, which, given that the structure of the input data is that of continuous temporal sequences, gives them a distinct advantage. Though temporal sequentiality is implicitly taken into account by both ML techniques through extraterrestrial solar radiation, only the RNNs have an explicit perception of it [22,24], and thus they can capture complex temporal dependencies that the RF models cannot.

As for the equation methods, the T_{dew} iterations fared better than their T_{min} counterparts in every iteration and almost every metric, indicating that Valiantzas’s [14] method is a better alternative to the Hargreaves and Samani [7] method. The iterations where k_{Rs} was calibrated on the totality of the dataset performed significantly worse than the other two use cases, and the non-calibrated iteration of the T_{dew} equation was the top performer among the equation methods, even against its locally calibrated counterpart. This could be explained by the fact that by performing the same calibration procedures on the validation dataset, the discovered “optimal” k_{Rs} values for the T_{dew} iterations in this specific timeframe were generally numerically closer to the commonly accepted 0.17 than the values derived from calibrating k_{Rs} on the training dataset (Table 6). The locally calibrated T_{min} equation, however, was the top performer among the T_{min} iterations, suggesting that calibration based on location-specific historical data can be a viable approach, though its accuracy seems to involve a degree of randomness.

Table 6. Calibrated k_{RS} values for all stations.

Station	Local k_{RS} Eq.4	Total k_{RS} Eq. 4	Local k_{RS} Eq.5	Total k_{RS} Eq. 5
AUA 01	0.155		0.154	
AUA 02	0.152		0.148	
AUA 03	0.150		0.138	
AUA 04	0.135		0.129	
AUA 05	0.127		0.135	
AUA 06	0.130	0.143	0.135	0.137
AUA 07	0.133		0.121	
AUA 08	0.170		0.145	
AUA 09	0.120		0.120	
AUA 10	0.168		0.151	

Table 7. Root Mean Square Error values (in Wm^{-2}) for each station and total.

	AUA 01	AUA 02	AUA 03	AUA 04	AUA 05	AUA 06	AUA 07	AUA 08	AUA 09	AUA 10	Total
Local Equation T_{min}	54.27	46.29	48.50	43.83	44.62	45.20	39.50	47.37	76.26	65.39	51.10
Local Equation T_{dew}	54.68	43.92	48.11	44.87	45.98	46.12	38.81	47.33	76.92	60.73	50.85
Total Equation T_{min}	58.67	49.30	51.07	40.51	43.41	41.94	36.65	61.36	50.68	90.56	54.12
Total Equation T_{dew}	61.33	49.74	48.63	40.25	44.96	44.96	32.83	53.17	54.63	84.38	53.00
Non-calibrated Equation T_{min}	54.11	47.97	48.33	47.00	63.86	55.95	46.65	47.37	50.65	64.20	53.18
Non-calibrated Equation T_{dew}	55.45	43.91	47.10	47.27	49.65	46.07	52.96	47.48	44.89	49.65	48.75
RFcomplete	56.65	45.40	43.97	38.98	43.15	40.91	31.33	47.42	50.86	65.41	47.39
RFhalf	57.40	46.07	44.64	38.60	43.78	41.66	31.69	48.53	51.08	66.08	47.93
RFminimal	60.65	49.63	48.82	42.19	49.01	46.33	35.78	51.74	51.33	72.21	51.71
ANNcomplete	46.69	33.98	36.59	32.99	35.90	33.18	36.83	39.86	53.67	51.95	41.25
ANNhalf	49.50	37.23	37.87	34.97	38.26	35.90	35.28	43.90	53.31	57.75	43.52
ANNminimal	52.62	43.76	45.71	36.40	41.71	38.28	37.67	44.58	68.41	58.50	48.20
Legend	Best Performance					Worst Performance					

A closer look at the models' performance for each station (Table 7) provides some interesting insights. The ANNcomplete model was the top performer for 7 out of 10 locations, in addition to being the best performer on the total dataset and the second best performer on AUA10. However, for the two remaining stations it ranked on the lower half of performers with significantly lower scores than the top performers; the ANNhalf model performed slightly better. This could potentially arise from the relationships between the selected variables in the majority of stations being roughly equivalent but significantly different than those same relationships in the remaining stations, creating bias [36]. Thus, an interesting venue for the future would be the comparison of equivalent ML models with the integration of explicit spatial awareness features and ML models trained exclusively in and for each location.

The three stations where the ANNcomplete model didn't rank at the top were AUA07, AUA09, and AUA10. The former was dominated by the RFcomplete model, closely followed by the RFhalf model, and with RFminimal also ranking within the top half of performers. As for the latter two locations, the non-calibrated T_{dew} equations achieved the highest accuracy in AUA09, specifically by a significant margin. As mentioned before, in AUA10, the ANNcomplete model ranked second, but it is important to mention that ANNhalf and ANNminimal ranked third and fourth best in terms of accuracy.

Invariably, the worst performer spot in every location was held by some iteration of the equation methods. Somewhat unsurprisingly, the totally calibrated T_{min} equation ranked as the worst performer in three locations, and its T_{dew} counterpart in two locations. The locally calibrated T_{dew} equation was the worst performer in AUA09, closely following its T_{min} counterpart, both having a significant difference from the third worst rank held by ANNminimal. Last, despite the non-calibrated T_{dew} equation being the top performer in two locations, it ranked as the worst performer in the other two stations, mirroring its T_{min} counterpart. The aforementioned observations seem to imply that simultaneous calibration for multiple locations within an area as large and as geomorphologically diverse as Nemea is ill-advised. Furthermore, it seems to vindicate Samani's recommendation of calibrating k_{RS} based on monthly temperature ranges for each specific location [17] as opposed to calibrations based on historical site-specific data.

Regarding site-specific bias (Table 8), there seems to be a stable scale between the sites for the relative MBE values for each method. Methods that consistently overestimate will have comparatively higher values to other methods even if their scores are negative for that specific location, and vice versa. Expectedly, this does not hold true for the locally calibrated equations, as they were independently calibrated for each station, as opposed to the other methods that were invariably optimized for the totality of locations. This sliding scale of bias further supports the idea of incorporating explicit spatial awareness for potential follow-up ML models. Additional information on all statistical measures is provided in the Appendix Table A1 to A3.

Table 8. Mean Bias Error values (in Wm⁻²) for each station and total.

	AUA01	AUA02	AUA03	AUA04	AUA05	AUA06	AUA07	AUA08	AUA09	AUA10	Total
Local Equation T _{min}	-13.23	-7.52	-13.11	-13.87	-10.80	-11.82	-9.60	-5.06	-39.94	-26.76	-15.17
Local Equation T _{dew}	-14.17	-14.42	-22.77	-21.35	-17.58	-18.98	-16.20	-17.76	-47.27	-30.54	-22.06
Total Equation T _{min}	-26.63	-17.60	-20.47	-4.58	10.31	4.37	1.31	-31.85	-12.42	-52.97	0.19
Total Equation T _{dew}	-33.17	-26.56	-23.84	-12.09	-15.20	-16.69	2.11	-24.42	-27.96	-46.49	-0.95
Non-calibrated Equation T _{min}	3.51	12.64	7.92	26.76	45.94	38.22	30.76	-5.06	19.77	-24.68	15.89
Non-calibrated Equation T _{dew}	3.71	9.87	11.57	26.08	24.11	21.37	39.88	11.29	9.32	-9.04	14.94
RFcomplete	-32.22	-27.46	-24.70	-15.02	-18.97	-15.70	4.56	-25.28	-25.70	-48.23	-22.64
RFhalf	-33.01	-27.99	-24.56	-14.17	-18.88	-15.69	4.16	-26.20	-26.02	-48.85	-22.88
RFminimal	-29.47	-21.18	-19.75	-6.13	-5.67	-5.23	4.14	-27.51	-18.52	-52.83	-17.84
ANNcomplete	-11.43	-3.71	3.93	6.80	-5.26	0.07	20.51	-9.93	12.13	-28.00	-1.19
ANNhalf	-16.22	-7.84	0.47	2.15	-9.68	-3.92	17.73	-13.38	-2.33	-32.61	-6.28
ANNminimal	-15.03	-7.27	5.15	2.68	-12.41	-5.57	19.58	-9.78	-18.47	-31.54	-7.04
Legend	Relative High Bias Score										Relative Low Bias Score

4. Conclusions

Our aim of this paper was to examine various methods for the purposes of filling in blank or erroneous solar radiation readings in the event of sensor malfunction and in different regimes of substitute weather parameters availability. Considering only sheer accuracy with a generalized approach, the ANNcomplete model is evidently the best pick, with the equation methods showing, on average, worse performance compared to the average performance of the ML models. However, to properly select the most appropriate of the examined methods for a given application, they should be compared in groups based on required input parameters. The “complete” ML models are only comparable to one another since they both use the totality of available routine meteorological data and in that regard, the ANNcomplete model is a better option with the exception of one single location. The “half” ML models are directly comparable to Eq.5, as they both utilize the same set of predictor variables, though in different forms. In most cases, again, the RNNs seem to be the best pick, though they lose out against RF in the same location as their “complete” counterparts. In two locations, the non-calibrated Tdew performed better than the respective ML methods. Finally, the “minimal” variants are comparable to the Tmin equations, as all require the absolute minimum amount of necessary predictor variables. In this category, the picture painted is almost identical to the aforementioned category, albeit with absolutely lower performance.

Additionally, the relative difficulty of development and implementation of each method should be taken into account. The equation methods are the most readily available to implement, requiring no specialized software or hardware to utilize and directly offering results of acceptable accuracy without requiring a massive preparation timespan. The ML methods, on the other hand, require specialized software, a moderate degree of technical knowledge, and relatively robust machines to both develop and implement, making them less accessible. Once developed, however, any given ML model can be deployed for any relevant application with ease, requiring roughly equivalent preparation effort and time to the empirical methods. Between them, RNNs required the largest amount of development time and processing power by a non-negligible margin. In our experience during this application, given the same number of input variables, time steps, and tunable hyperparameter combinations, RNN models would take about 70–150% longer to train than their equivalent RF models, not to mention that they are also considerably less straightforward to implement and significantly harder to interpret [37].

Given our findings, the selection of the most appropriate method is a multifaceted process with no concrete universal answer. Given sufficient resources and granted a focus on sheer accuracy, RNNs are the best choice, but they are also the most demanding of the methods examined. Random Forest’s performance did not constitute a significant enough improvement to the equations’ to be considered a viable option in most cases. The empirical methods proved competitive against the ML models, and their deficiencies in accuracy compared to the equivalent ML models can be made up for by their relative ease of use. It is also important to mention that the equations examined are spatially generalized in their function, while the trained ML models are locally restricted to the study area and are expected to underperform in different locations; even within this one study area, their performance varied significantly across sites. It is our hope that this study will serve as a useful resource to inform interested parties in deciding the most cost-effective methods to solve the problem of invalid solar radiation measurements according to their needs and capabilities.

Researchers could focus on examining the viability of location-specific ML models for the express purpose of filling in data gaps in known station locations or the possibility of creating more generalized models through the integration of explicit spatial and temporal parameters, which, in

addition to being used for the aforementioned purpose, they could find applications in regressing solar radiation values for stations that do not normally record them.

Author Contributions

Soulis K, Conceptualization, Methodology, Data curation, Supervision, Coding, Resources, Review & Editing, Nikitakis E, Coding, Methodology, Data curation, Writing, Validation, Investigation, Methodology, Visualization, Formal analysis, Katsogiannou A, Data curation, Coding, Kalivas D, Resources, Supervision

Acknowledgements

The authors would like to thank the anonymous reviewers for their contributions and constructive criticism.

Funding

The installation of the meteorological stations used in this study was co-funded by the NSRF and the European Union. This work was supported by DT-Agro project, grand number 014815 that is carried out within the framework of the National Recovery and Resilience Plan Greece 2.0, funded by the European Union—NextGenerationEU (Implementation body: HFRI), <https://greece20.gov.gr>.

Use of AI tools declaration

The authors declare they have not used Artificial Intelligence (AI) tools in the creation of this article.

Data availability

The data used in this study can be made available upon reasonable request.

Conflicts of interest

The authors hereby declare no conflict of interest.

References

1. Colle S, De Abreu SL, Ruther R (2001) Uncertainty in economic analysis of solar water heating and photovoltaic systems. *Sol Energy* 70: 131–142. [https://doi.org/10.1016/S0038-092X\(00\)00134-1](https://doi.org/10.1016/S0038-092X(00)00134-1)
2. Allen RG, Pereira LS, Raes D, et al. (1998) Crop evapotranspiration: guidelines for computing crop water requirements. Available from FAO eBooks (Issue 1). Available from: <https://www.fao.org/4/x0490e/x0490e00.htm>.
3. Zang H, Xu Q, Bian H (2012) Generation of typical solar radiation data for different climates of China. *Energy* 38: 236–248. <https://doi.org/10.1016/j.energy.2011.12.008>
4. Zang H, Jiang X, Cheng L, et al. (2022) Combined empirical and machine learning modeling method for estimation of daily global solar radiation for general meteorological observation stations. *Renew. Energy* 195: 795–808. <https://doi.org/10.1016/j.renene.2022.06.063>

5. Ağbulut Ü, Gürel AE, Biçen Y (2021) Prediction of daily global solar radiation using different machine learning algorithms: Evaluation and comparison. *Renewable Sustainable Energy Rev* 135: 110114. <https://doi.org/10.1016/j.rser.2020.110114>
6. Soulis K, Kalivas D, Apostolopoulos C (2018) Delimitation of agricultural areas with natural constraints in Greece: Assessment of the dryness climatic criterion using geostatistics. *Agronomy* 8: 161. <https://doi.org/10.3390/agronomy8090161>
7. Hargreaves GH, Samani ZA (1982) Estimating potential evapotranspiration. *J Irrig Drain Div* 108: 225–230. <https://doi.org/10.1061/jrcea4.0001390>
8. Meza FJ, Yebra ML (2016) Estimation of daily global solar radiation as a function of routine meteorological data in Mediterranean areas. *Theor Appl Climatol* 125: 479–488. <https://doi.org/10.1007/s00704-015-1519-6>
9. Mousavi SM, Mostafavi ES, Jaafari A, et al. (2015) Using measured daily meteorological parameters to predict daily solar radiation. *Measurement* 76: 148–155. <https://doi.org/10.1016/j.measurement.2015.08.004>
10. Thota SKR, Mala C, Chandamuri P, et al. (2023) Solar Radiation Prediction Using the Random Forest Regression Algorithm. In: Haldorai A, Ramu A, Mohanram S, et al. Eds., 4th EAI International Conference on Big Data Innovation for Sustainable Cognitive Computing. EAI/Springer Innovations in Communication and Computing. Springer, Cham. https://doi.org/10.1007/978-3-031-07654-1_11
11. Villegas-Mier C, Rodriguez-Resendiz J, Álvarez-Alvarado J, et al. (2022) Optimized random forest for solar radiation prediction using sunshine hours. *Micromachines* 13: 1406. <https://doi.org/10.3390/mi13091406>
12. Taki M, Rohani A, Yildizhan H (2021) Application of machine learning for solar radiation modeling. *Theor Appl Climatol* 143: 1599–1613. <https://doi.org/10.1007/s00704-020-03484-x>
13. Demir V, Citakoglu H (2023) Forecasting of solar radiation using different machine learning approaches. *Neural Comput Applic* 35: 887–906. <https://doi.org/10.1007/s00521-022-07841-x>
14. Valiantzas JD (2013) Simplified forms for the standardized FAO-56 Penman–Monteith reference evapotranspiration using limited weather data. *J Hydrol* 505: 13–23. <https://doi.org/10.1016/j.jhydrol.2013.09.005>
15. Fernández-Delgado M, Cernadas E, Barro S, et al. (2014) Do we need hundreds of classifiers to solve real-world classification problems? *J Mach Learn Res* 15: 3133–3181.
16. LeCun Y, Bengio Y, Hinton G (2015) Deep learning. *Nature* 521: 436–444. <https://doi.org/10.1038/nature14539>
17. Samani Z (2004) Discussion of “History and Evaluation of Hargreaves Evapotranspiration Equation” by George H. Hargreaves and Richard G. Allen. *J Irrig Drain Eng* 130: 447–448. [https://doi.org/10.1061/\(ASCE\)0733-9437\(2004\)130:5\(447.2\)](https://doi.org/10.1061/(ASCE)0733-9437(2004)130:5(447.2))
18. Montgomery DC, Peck EA, Vining GG (2012) *Introduction to linear regression analysis*, 5th Ed., Hoboken: John Wiley & Sons.
19. Breiman L (2001) Random forests. *Mach Learn* 45: 5–32. <https://doi.org/10.1023/A:1010933404324>
20. Hastie T (2009) *The elements of statistical learning: data mining, inference, and prediction*. 2nd Ed., New York: Springer Science & Business Media.
21. Probst P, Wright MN, Boulesteix AL (2019) Hyperparameters and tuning strategies for random forest. *Wiley Interdiscip Rev Data Min Knowl Disc* 9: e1301. <https://doi.org/10.1002/widm.1301>

22. Oshiro TM, Perez PS, Baranauskas JA (2012) How Many Trees in a Random Forest? In: Perner P, Eds., *Machine Learning and Data Mining in Pattern Recognition*. MLDM 2012. Lecture Notes in Computer Science, 7376. Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-642-31537-4_13
23. Geurts P, Ernst D, Wehenkel L (2006) Extremely randomized trees. *Mach Learn* 63: 3–42. <https://doi.org/10.1007/s10994-006-6226-1>
24. Schmidhuber J (2015) Deep learning in neural networks: An overview. *Neural Networks* 61: 85–117. <https://doi.org/10.1016/j.neunet.2014.09.003>
25. Lipton ZC, Berkowitz J, Elkan C (2015) A critical review of recurrent neural networks for sequence learning. arXiv. <https://doi.org/10.48550/arxiv.1506.00019>
26. Cabello-Solorzano K, Ortigosa de Araujo I, Peña M, et al. (2023) The Impact of Data Normalization on the Accuracy of Machine Learning Algorithms: A Comparative Analysis. In: García Bringas P, et al. 18th International Conference on Soft Computing Models in Industrial and Environmental Applications (SOCO 2023). Lecture Notes in Networks and Systems, 750. Springer, Cham. https://doi.org/10.1007/978-3-031-42536-3_33
27. Sutskever I, Vinyals O, Le QV (2014) Sequence to sequence learning with neural networks. arXiv. <https://doi.org/10.48550/arXiv.1409.3215>
28. Hochreiter S, Schmidhuber J (1997) Long short-term memory. *Neural Comput* 9: 1735–1780. <https://doi.org/10.1162/neco.1997.9.8.1735>
29. Srivastava N, Hinton G, Krizhevsky A, et al. (2014) Dropout: A simple way to prevent neural networks from overfitting. *J Mach Learn Res* 15: 1929–1958.
30. Goodfellow I, Bengio Y, Courville A (2016) *Deep learning*, Cambridge: MIT Press. Available from: <https://www.deeplearningbook.org/>.
31. Willmott C, Matsuura K (2005) Advantages of the mean absolute error (MAE) over the root mean square error (RMSE) in assessing average model performance. *Clim Res* 30: 79–82. <https://doi.org/10.3354/cr030079>
32. Chai T, Draxler RR (2014) Root mean square error (RMSE) or mean absolute error (MAE)? — Arguments against avoiding RMSE in the literature. *Geosci Model Dev* 7: 1247–1250. <https://doi.org/10.5194/gmd-7-1247-2014>
33. Tymvios F, Jacovides C, Michaelides S, et al. (2005) Comparative study of Ångström's and artificial neural networks' methodologies in estimating global solar radiation. *Sol Energy* 78: 752–762. <https://doi.org/10.1016/j.solener.2004.09.007>
34. Louppe G (2015) Understanding random forests: from theory to practice. *Mach Learn*. Available from: <https://doi.org/10.48550/arXiv.1506.00019>
35. Cutler DR, Edwards TC, Beard KH, et al. (2007) Random forests for classification in ecology. *Ecology* 88: 2783–2792. <https://doi.org/10.1890/07-0539.1>
36. Marcus GF (2018) Deep learning: A critical appraisal. *Artif Intell*. <https://doi.org/10.48550/arxiv.1801.00631>
37. Molnar C (2020) Interpretable machine learning: A guide for making black box models explainable. Christoph Molnar. Available from: <https://christophm.github.io/interpretable-ml-book/>.



AIMS Press

© 2024 the Author(s), licensee AIMS Press. This is an open access article distributed under the terms of the Creative Commons Attribution License (<http://creativecommons.org/licenses/by/4.0>)

Appendix

Additional Metric Tables

Table A1. R² for each station and total.

	AUA0 1	AUA0 2	AUA0 3	AUA0 4	AUA0 5	AUA0 6	AUA0 7	AUA0 8	AUA0 9	AUA1 0	Total
Local Equation T _{min}	0.617	0.735	0.671	0.774	0.755	0.760	0.794	0.766	0.552	0.673	0.710
Local Equation T _{dew}	0.612	0.761	0.676	0.763	0.740	0.750	0.801	0.766	0.521	0.718	0.713
Total Equation T _{min}	0.553	0.700	0.635	0.807	0.768	0.793	0.823	0.618	0.745	0.374	0.674
Total Equation T _{dew}	0.511	0.694	0.669	0.809	0.751	0.762	0.858	0.713	0.704	0.457	0.687
Non- calibrated Equation T _{min}	0.620	0.716	0.673	0.740	0.498	0.632	0.713	0.766	0.745	0.685	0.686
Non- calibrated Equation T _{dew}	0.600	0.762	0.690	0.737	0.697	0.750	0.630	0.765	0.800	0.812	0.736
RFcomplete	0.583	0.745	0.730	0.821	0.771	0.803	0.871	0.744	0.742	0.618	0.746
RFhalf	0.572	0.738	0.721	0.825	0.764	0.796	0.868	0.732	0.739	0.610	0.740
RFminimal	0.522	0.695	0.667	0.791	0.704	0.747	0.831	0.695	0.737	0.534	0.697
ANNcompl ete	0.717	0.857	0.813	0.872	0.841	0.870	0.821	0.819	0.712	0.759	0.810
ANNhalf	0.682	0.829	0.799	0.856	0.820	0.848	0.836	0.780	0.716	0.702	0.789
ANNminim al	0.640	0.763	0.708	0.844	0.786	0.828	0.813	0.774	0.533	0.694	0.740
Legend	Best Performance						Worst Performance				

Table A2. relative Root Mean Square Error for each station and total.

	AUA	AUA	AUA	AUA	AUA	AUA	AUA	AUA	AUA	AUA	Total
	01	02	03	04	05	06	07	08	09	10	
Local	29.14	26.04	28.39	25.70	25.01	26.02	25.54	25.44	36.61	32.47	28.58
Equation	%	%	%	%	%	%	%	%	%	%	%
T_{\min}											
Local	29.36	24.71	28.16	26.31	25.78	26.55	25.09	25.41	37.84	30.15	28.45
Equation	%	%	%	%	%	%	%	%	%	%	%
T_{dew}											
Total	31.50	27.73	29.89	23.75	24.34	24.15	23.69	32.00	27.62	44.84	30.30
Equation	%	%	%	%	%	%	%	%	%	%	%
T_{\min}											
Total	32.93	27.98	28.47	23.60	25.21	25.89	21.22	27.73	29.77	41.78	29.67
Equation	%	%	%	%	%	%	%	%	%	%	%
T_{dew}											
Non-	29.05	26.98	28.29	27.56	35.80	32.21	30.16	25.44	27.62	31.88	29.75
calibrated	%	%	%	%	%	%	%	%	%	%	%
Equation											
T_{\min}											
Non-	29.77	24.70	27.57	27.71	27.83	26.52	34.23	25.50	24.47	24.65	27.26
calibrated	%	%	%	%	%	%	%	%	%	%	%
Equation											
T_{dew}											
RFcomplete	30.41	25.54	25.74	22.85	24.19	23.55	20.25	26.60	27.91	30.66	26.42
	%	%	%	%	%	%	%	%	%	%	%
RFhalf	30.82	25.92	26.13	22.63	24.54	23.98	20.49	27.23	28.03	30.97	26.73
	%	%	%	%	%	%	%	%	%	%	%
RFminimal	32.56	27.92	28.58	24.73	27.48	26.67	23.13	29.03	28.17	33.84	28.84
	%	%	%	%	%	%	%	%	%	%	%
ANNcomplete	25.07	19.12	21.42	19.34	20.13	19.10	23.81	22.36	29.45	24.35	22.84
	%	%	%	%	%	%	%	%	%	%	%
ANNhalf	26.58	20.94	22.17	20.50	21.45	20.67	22.80	24.63	29.25	27.06	24.10
	%	%	%	%	%	%	%	%	%	%	%
ANNminimal	28.25	24.62	26.76	21.34	23.39	22.04	24.35	25.01	37.54	27.42	26.73
	%	%	%	%	%	%	%	%	%	%	%
Legend	Best Performance						Worst Performance				

Table A3. Mean Absolute Error for each station and total.

	AUA0 1	AUA0 2	AUA0 3	AUA0 4	AUA0 5	AUA0 6	AUA0 7	AUA0 8	AUA0 9	AUA1 0	Tota 1
Local Equation T_{\min}	45.53	37.13	40.34	35.22	36.05	36.04	31.35	38.45	52.42	54.71	40.6 7
Local Equation T_{dew}	45.15	36.54	40.19	37.00	37.99	38.48	31.61	39.59	56.05	51.61	41.3 7
Total Equation T_{\min}	50.02	40.08	42.74	31.89	32.03	31.56	28.43	52.10	37.90	76.49	42.0 2
Total Equation T_{dew}	51.66	41.45	40.66	32.59	37.01	37.38	25.17	45.04	42.63	69.45	42.1 4
Non- calibrated Equation T_{\min}	43.46	35.97	37.83	34.02	48.80	41.93	35.24	38.45	35.39	53.58	40.4 4
Non- calibrated Equation T_{dew}	44.09	33.47	36.74	33.60	35.88	32.17	41.16	36.23	31.08	40.68	36.5 1
RFcomplete	46.95	38.10	36.51	31.53	35.87	33.35	23.02	41.37	39.28	55.38	37.9 6
RFhalf	47.68	38.59	37.16	31.02	36.34	33.87	23.26	42.26	39.65	55.84	38.3 9
RFminimal	51.17	40.72	40.24	33.99	39.25	36.80	26.15	45.63	39.76	61.30	41.3 0
ANNcomplete	37.59	27.26	29.24	24.62	26.40	24.38	27.41	33.56	39.14	42.06	31.0 5
ANNhalf	40.68	30.48	30.90	27.18	29.75	27.33	26.50	37.79	39.17	47.24	33.5 5
ANNminimal	44.22	35.22	36.62	28.00	33.16	29.27	27.57	38.89	50.13	48.33	37.0 3
Legend	Best Performance						Worst Performance				

R Packages Used in this paper

- base R Core Team (2024) R: A Language and Environment for Statistical Computing. (R version 4.4.1) R Foundation for Statistical Computing, Vienna, Austria. <https://www.R-project.org/>.
- keras3 Kalinowski T, Allaire J, Chollet F (2024) keras3: R Interface to “Keras”. R package version 1.2.0, <https://CRAN.R-project.org/package=keras3>.

- tensorflow Allaire J, Tang Y (2024) tensorflow: R Interface to “TensorFlow”. R package version 2.16.0, <https://CRAN.R-project.org/package=tensorflow>.
- caret Kuhn M (2008) Building Predictive Models in R Using the caret Package. *J Stat Software* 28: 1–26. <https://doi.org/10.18637/jss.v028.i05>
- ranger Wright MN, Ziegler A (2017) ranger: A Fast Implementation of Random Forests for High Dimensional Data in C++ and R. *J Stat Software* 77: 1–17. <https://doi.org/10.18637/jss.v077.i01>
- randomForest Liaw A, Wiener M (2002) “Classification and Regression by randomForest.” *R News*, *2*(3), 18–22. <https://CRAN.R-project.org/doc/Rnews/>.
- readxl Wickham H, Bryan J (2023) readxl: Read Excel Files. R package version 1.4.3, <https://CRAN.R-project.org/package=readxl>.
- openxlsx Schauburger P, Walker A (2024) openxlsx: Read, Write and Edit xlsx Files. R package version 4.2.7.1, <https://CRAN.R-project.org/package=openxlsx>.
- ggplot H. Wickham. ggplot2: Elegant Graphics for Data Analysis. Springer-Verlag New York, 2016.
- gridExtra Auguie B (2017) gridExtra: Miscellaneous Functions for “Grid” Graphics. R package version 2.3, <https://CRAN.R-project.org/package=gridExtra>.
- writexl Ooms J (2024) writexl: Export Data Frames to Excel “xlsx” Format. R package version 1.5.0, <https://CRAN.R-project.org/package=writexl>.
- dplyr Wickham H, François R, Henry L, et al. (2023) dplyr: A Grammar of Data Manipulation. R package version 1.1.4, <https://CRAN.R-project.org/package=dplyr>.
- moments Komsta L, Novomestky F (2022) moments: Moments, Cumulants, Skewness, Kurtosis and Related Tests. R package version 0.14.1, <https://CRAN.R-project.org/package=moments>.