Solar Irradiance Probabilistic Forecasting Using Machine Learning, Metaheuristic Models and Numerical Weather Predictions

Vateanui Sansine, Pascal Ortega, Daniel Hissel, Marania Hopuare

sustainability

MDPI

Article

Solar Irradiance Probabilistic Forecasting Using Machine Learning, Metaheuristic Models and Numerical Weather Predictions

Vateanui Sansine [1,2,*], Pascal Ortega [1], Daniel Hissel [2] and Marania Hopuare [1]

[1] GEPASUD, Université de Polynésie Française, Campus d'Outumaoro, 98718 Puna'auia, Tahiti, French Polynesia
[2] FEMTO-ST/FCLAB, Université de Franche-Comté, CNRS, Rue Thierry Meg, CEDEX, F-90010 Belfort, France
* Correspondence: vateanui.sansine@doctorant.upf.pf; Tel.: +689-40-80-38-76

Abstract: Solar-power-generation forecasting tools are essential for microgrid stability, operation, and planning. The prediction of solar irradiance (SI) usually relies on the time series of SI and other meteorological data. In this study, the considered microgrid was a combined cold- and power-generation system, located in Tahiti. Point forecasts were obtained using a particle swarm optimization (PSO) algorithm combined with three stand-alone models: XGboost (PSO-XGboost), the long short-term memory neural network (PSO-LSTM), and the gradient boosting regression algorithm (PSO-GBRT). The implemented daily SI forecasts relied on an hourly time-step. The input data were composed of outputs from the numerical forecasting model AROME (Météo France) combined with historical meteorological data. Our three hybrid models were compared with other stand-alone models, namely, artificial neural network (ANN), convolutional neural network (CNN), random forest (RF), LSTM, GBRT, and XGboost. The probabilistic forecasts were obtained by mapping the quantiles of the hourly residuals, which enabled the computation of 38%, 68%, 95%, and 99% prediction intervals (PIs). The experimental results showed that PSO-LSTM had the best accuracy for day-ahead solar irradiance forecasting compared with the other benchmark models, through overall deterministic and probabilistic metrics.

Keywords: solar irradiance; forecasting; numerical weather predictions; machine learning; deep learning; metaheuristic models; optimization

Citation: Sansine, V.; Ortega, P.; Hissel, D.; Hopuare, M. Solar Irradiance Probabilistic Forecasting Using Machine Learning, Metaheuristic Models and Numerical Weather Predictions. *Sustainability* **2022**, *14*, 15260. https://doi.org/10.3390/su142215260

Academic Editors: Luis Hernández-Callejo, Sergio Nesmachnow and Sara Gallardo Saavedra

Received: 9 October 2022
Accepted: 31 October 2022
Published: 17 November 2022

1. Introduction

Global electricity demand is expected to rise by 2.4% in 2022, despite economic weaknesses and high prices [1]. This rise, driven by the growth of the world population, the industrialization of developing countries, and the worldwide process of urbanization [2], uses fossil fuels as the main power source. This has proven to be detrimental for the environment and the climate. Therefore, renewable energies have gained a lot of attention, especially photovoltaics (PVs), due to their accessibility, low cost, lifetime, and environmental benefits. Solar PV installations are growing faster than any other renewable energy. Indeed, PVs are forecast to account for 60% of the increase in global renewable capacity in 2022 [3]. In this context, PVs provide many environmental and economic benefits. However, uncontrollable factors such as the weather, seasonality, and climate lead to intermittent, random, and volatile PV power generation. These significant constraints still hinder the large-scale integration of PVs into the power grid and interfere with the reliability and stability of existing grid-connected power systems [4]. Thus, a reliable forecast of PV power outputs is essential to ensure the stability, reliability, and cost-effectiveness of the system [5]. Those forecasts are usually implemented through prediction of the global horizontal irradiance (GHI). There are three main groups of solar irradiance forecasting model [6]:

- Statistical models, which are based on historical data and their ability to extract information/patterns from the data to forecast time series.
- Physical models, which are based on sky images, satellite images, or numerical weather predictions (NWPs) to infer the dynamics of solar radiation through the atmosphere.
- Hybrid models, which exploit statistical and physical models to obtain forecasts with higher precision.

Machine learning algorithms, classified under statistical models, have become very popular for studies related to PV power-output forecasting, and play an important role in contemporary solar-irradiance forecasting for conventional grid management and for smaller and independent microgrids.

Ogliari et al. [7] compared two deterministic models with hybrid methods, a combination of an artificial neural network (ANN) and a clear sky radiation model, for PV power output forecasting. The models were trained on one year of measured data in a PV plant located in Milan, Italy. The results show that the hybrid method is the most precise for PV output forecasting, demonstrating advantages by combining physical models with machine learning algorithms.

Crisosto et al. [8] used a feedforward neural network (FFNN) with Levenberg–Marquardt backpropagation (LM–BP) to make predictions for one hour ahead with one-minute resolution in the city of Hanover, Germany. The model was trained on a four-year dataset including all-sky images, used for cloud cover computation, and measured global irradiance. For hourly average predictions, the FFNN-LM-BP showed the best results with an RMSE (Wh/m^2) = 65, and R^2 = 0.98, compared with the persistence model with an RMSE = 91 and R^2 = 0.91.

Yu et al. [9] used a long short-term memory (LSTM) model to predict GHI in three cities in the USA, namely, New York, Atlanta, and Hawaii. The time horizons of the model were one hour ahead and one day ahead. The model's performance was compared with other models such as the autoregressive integrated moving average (ARIMA), convolutional neural network (CNN), FFNN, and recurrent neural network (RNN). For hourly predictions, the LSTM model was more precise in all three states, with R^2 exceeding 0.9 on cloudy and partially cloudy days, whereas R^2 for the RNN was only 0.70 and 0.79 in Atlanta and Hawaii. For daily forecasting, LSTM outperformed the other models except in clear-sky days for New York, whereas for Hawaii and Atlanta, LSTM was better in every case.

However, it is difficult to improve the forecast from only one machine learning model, which sometimes suffers from instability originating from poor parameter choice, or from a reduced number of input variables. Ensemble learning is a popular development trend in artificial intelligence (AI) algorithms [10]. It combines independent models with stronger learners, which can achieve better stability and prediction effects compared with individual models [11].

Huang et al. [12] used gradient boosting regression (GBRT), extreme gradient lifting (XGboost), Gaussian process regression (GPR), and random forest (RF) models to carry out GHI predictions. Those ensemble models performed better than other stand-alone models such as decision tree (DT), backpropagation neural network (BPNN), and support vector machine regression (SVR). It is concluded that the stacking models—including GBRT, XGboost, GPR, and RF—are the best models to predict solar radiation.

Li et al. [13] used XGboost to implement point forecasts for solar irradiance and kernel density estimation (KDE) to generate probabilistic forecasts from the above prediction results. This method enabled the computation of confidence levels and demonstrated better results than other benchmark algorithms such as SVR and random forest.

To improve the efficiency of machine learning (ML) models, an increasing number of studies have used metaheuristic models in order to optimize the parameters of the considered GHI forecasting model.

Jia et al. [14] utilized particle swarm optimization (PSO) coupled with a Gaussian exponential model (GEM) to predict daily and monthly solar radiation (Rs). The hybrid PSO-GEM model showed the best results for Rs prediction.

Duan et al. [15] used NWP, together with the kernel-based nonlinear extension of Arps decline (KNEA) to predict solar irradiance. The KNEA algorithm is optimized by a metaheuristic algorithm called the Bat algorithm (BA). The proposed method for GHI forecasting is called the BA-KNEA. Duan et al. also implemented other hybrid models such as PSO-XGboost, BA-XGboost, and PSO-KNEA. The results showed that BA-KNEA is better at performing solar radiation forecasts.

In summary, ensemble learning models are an emerging trend in ML, proving to be appropriate tools for regression, and therefore, GHI forecasting. They have shown good results compared with deep learning models for day-ahead GHI point forecasts [12]. Moreover, ensemble methods can be further improved with metaheuristic models for parameter optimization, as well as the prevention of potential numerical instability from which various ML models suffer. However, one of the drawbacks of point forecasts is that they contain limited information about the volatility and randomness of solar irradiance. Point forecasts cannot satisfy the needs of a power system's optimized operation [13]. For this reason, considerable attention has been drawn to probabilistic forecasting, which enables the computation of prediction intervals to provide to grid dispatchers in order to facilitate grid operation.

This study focused on the implementation of daily probabilistic forecasts with hybrid models such as PSO-XGboost, PSO-LSTM, PSO-GBRT, and quantile mapping for the computation of prediction intervals. The hybrid models were compared with other reference models, namely, ANN, CNN, LSTM, RF, and GBRT.

The novelty of this work lies in the residual modeling implemented with an innovative hybrid model (PSO-LSTM), enabling us to compute prediction intervals with different confidence levels, and thus obtain probabilistic forecasts. To the best of our knowledge, no day-ahead probabilistic GHI predictions have been implemented with this method. Secondly, we demonstrate that using a deep learning approach combined with metaheuristic models can achieve higher accuracy than ensemble models, or their optimized versions.

In order to produce those forecasts, historical data measured on-site coupled with NWP were used in the training of GHI forecasting models. These forecasting tools are intended to control a combined cold- and power-generation system, comprising several energy production and storage sub-systems, the whole being powered by solar energy. This prototype is called RECIF (the French abbreviation for a microgrid for electricity and cold cogeneration), and has been developed within the framework of a project funded by the French National Agency for Research (ANR) and is being implemented at the University of French Polynesia (UPF).

The rest of the paper is organized as follows: the historical data and the implemented data processes are presented in Section 2, followed in Section 3 by a theoretical background of machine learning and metaheuristic models. The results, analysis, suggestions for future research, and perspectives are presented in Section 4. The conclusions and the principal results are presented in Section 5.

2. Materials and Methods

2.1. Input Variables

This study utilized historical data measured from the weather station set-up in the University of French Polynesia. Two years of measurements are at our disposal, from 2019 to 2020. Those measurements are crucial in the design and implementation of a reliable forecasting system based on machine learning algorithms. The meteorological variables are measured with a time step of 1 min. The GHI is measured with a BF5 pyranometer supplied by Delta Devices, which uses an array of photodiodes with a unique computer-generated shading pattern to measure the diffuse horizontal irradiance (DHI) and GHI [16]. This enables the computation of the direct normal irradiance (DNI) for a given solar zenith angle. The set of inputs chosen from the weather station, for the GHI day-ahead forecasting models, was as follows:

- Ambient temperature T (°C);
- Dew point temperature (°C);
- Relative humidity H (%);
- Atmospheric pressure (hPa);
- Wind velocity *WV* (m/s) and the wind direction WD (°);
- Amount of rain (mm);
- Solar irradiance variables such as GHI, DHI, and DNI;
- The clear-sky model, GHI_{CLS}, as the theoretical value of GHI in clear-sky conditions.

An overview of the data is presented in Table 1. The processing of the historical data is detailed in Section 2.2.

Table 1. Descriptive statistics including the mean, standard deviation (std), minimum/maximum values, and the quantiles for each meteorological variable.

	GHI	DHI	DNI	Temperature	Rel Humidity	Pressure	Dew Point	Wind Speed	Wind Direction	Rain	CLS
Count	898,460	898,460	898,460	898,460	898,460	898,460	898,460	898,460	898,460	898,460	898,460
Mean	219.1	80.0	139.2	26.1	76.8	1005.5	21.6	1.8	133.1	0.0	297.9
Std	320.5	125.2	265.1	2.3	9.2	2.4	1.8	1.2	93.2	0.0	381.0
Min	0.2	0.2	0.0	18.4	35.0	996.0	12.1	0.0	0.0	0.0	0.0
25%	0.2	0.2	0.0	24.3	70.0	1004.0	20.5	1.1	68.8	0.0	0.0
50%	1.9	1.6	0.0	25.9	77.6	1006.0	21.8	1.7	110.3	0.0	0.0
75%	370.1	110.0	120.6	27.8	83.7	1007.0	22.9	2.3	214.2	0.0	649.0
Max	1253	814.1	1156.4	33.5	99.2	1013.0	26.8	13.4	360.0	2.8	1145.8

In addition to these in situ measurements, numerical weather predictions (NWPs) were used to train our day-ahead forecasting models. The numerical weather prediction model AROME was implemented by Météo-France with a resolution of 0.025 × 0.025° (2.5 × 2.5 km) in French Polynesia. These predictions have a maximum time horizon of 42 h and are updated every 12 h in French Polynesia. In Figure 1, each node (or grid point) of the AROME model for the north-eastern part of Tahiti is depicted, numbered from 1 to 34. Two years of NWP outputs are available, spanning from January 2019 to December 2020 with an hourly time-step. The GHI values predicted by AROME are only available from 9 am to 4 pm.

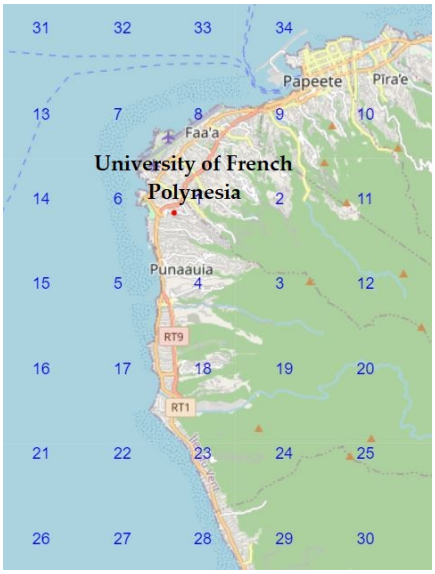

Figure 1. Points from the AROME grid (red dot represents the University).

2.2. Data Processing

This section explains the steps involved in processing the historical data and the AROME output. A vital step in data processing is to remove anomalous data that are

caused by technical glitches from the sensors, such as negative and Not a Number (or NaN) values, or outliers. After the removal of NaN values, the outliers are detected through the interquartile range method (IQR). A sliding mean is applied to the 1 min time-step meteorological data, in order to obtain hourly values, and making correlation with the AROME output possible. The mean value at time t is computed from the 60 previous measurements.

In order to quantify the errors between the in situ measurements and the AROME model, and then determine which points of the AROME grid to use for the training of the machine learning algorithms, the following metrics were used: the mean square error (MSE), the root-mean-square error (RMSE), and the determination coefficient (R^2).

$$\mathrm{MSE} = \frac{1}{N}\sum_{i=0}^{N}(y_{\mathrm{measured},i} - y_{\mathrm{predicted},i})^2, \quad (1)$$

$$\mathrm{RMSE} = \sqrt{\frac{1}{N}\sum_{i=0}^{N}(y_{\mathrm{measured},i} - y_{\mathrm{predicted},i})^2}, \quad (2)$$

$$R^2 = 1 - \frac{\sum_{i=0}^{N}(y_{\mathrm{measured},i} - y_{\mathrm{predicted},i})^2}{\sum_{i=0}^{N}(y_{\mathrm{measured},i} - \overline{y}_{\mathrm{measured}})^2}, \quad (3)$$

where N is the number of observations, $y_{\mathrm{measured},i}$ is the measurements, $\overline{y}_{\mathrm{measured}}$ is the mean value of the measurements, and $y_{\mathrm{predicted},i}$ is the predicted values. The results are presented in Figure 2.

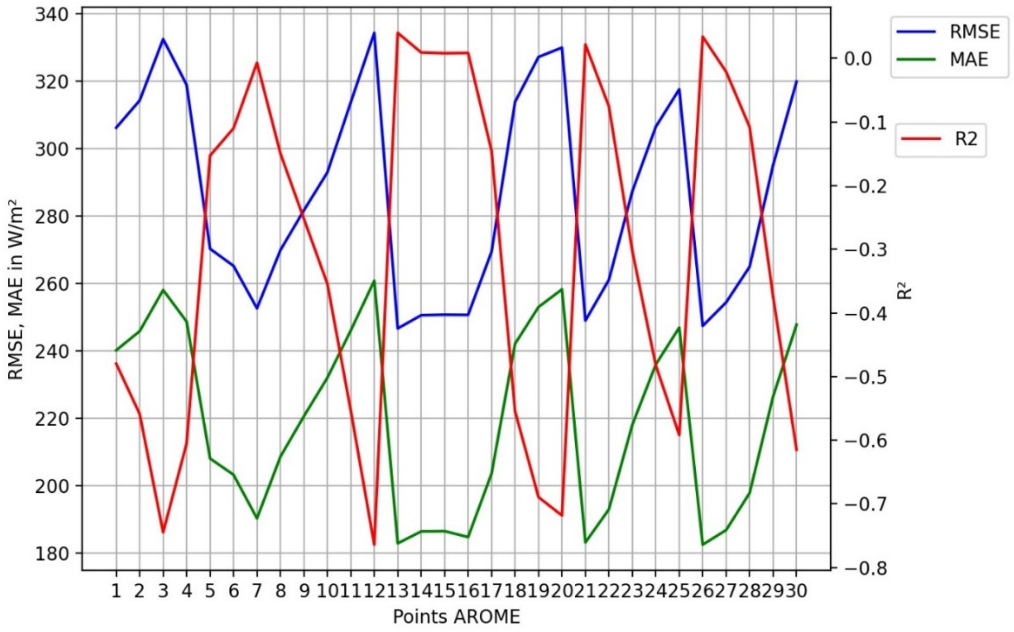

Figure 2. Errors between the measured GHI at the UPF and the AROME predictions for each grid point.

Points 31 to 34 were not used because they contained a great number of outliers in the first semester. The selected points were, arbitrarily, the points with some of the lowest correlation ($R^2 < -0.45$), i.e., n°3, 12, 20, 25, and points that exhibited positive correlation with the measured data, i.e., n°7, 13, 14, 15, 16, 17, 21, and 26.

The missing data were not replaced (through linear interpolation for example), but the consecutiveness of the dates of the data was ensured in the construction of the input data (or input vector); thus, no missing values were processed in the machine learning algorithms.

The night hours were removed from the measured data; consequently, the GHI forecasts implemented in this study were only performed for the hours between 6 am and 8 pm. After correlating the measurements and the AROME output, the merged data were normalized according to Equation (4):

$$X_{\text{normalized}} = \frac{X - X_{\text{min}}}{X_{\text{max}} - X_{\text{min}}}, \tag{4}$$

The data were then split into 70% as training data, 20% as validation data, and 10% as testing data.

3. Theoretical Background

3.1. Long Short-Term Memory (LSTM)

In recent years, LSTM has been widely applied to implement GHI forecasting [17,18]. One of the main advantages, compared with a classical RNN, is that LSTM models can deal with long-term dependencies found in the data without having problems such as vanishing gradients [19] using forget gates.

As shown in Figure 3, a typical LSTM network consists of one cell and three gates (an input gate, forget gate, and output gate). The input gate adjusts the amount of new data stored in the unit. The output gate determines which information to obtain from the cell, while the forget gate determines which information can be discarded [15]. Each gate uses either tanh or sigmoid as activation functions.

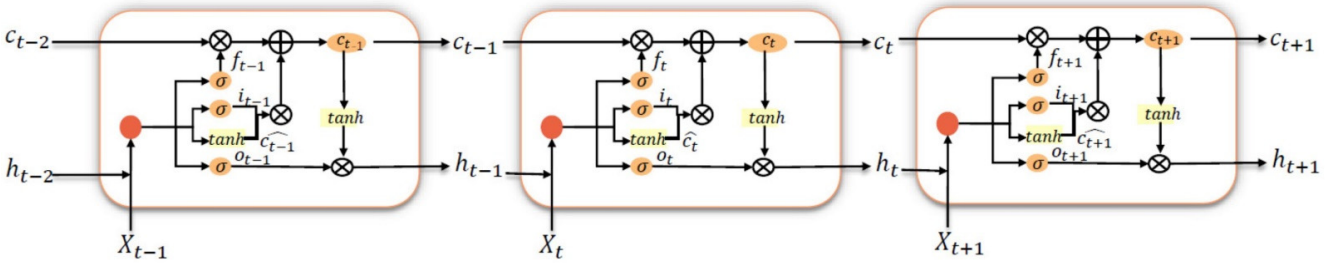

Figure 3. Basic structure of an LSTM model [15].

The input gate can be calculated with Equation (5) [15]:

$$gate(f_i) = \sigma_s(w_i x_t + u_i h_{t-1} + b_i), \tag{5}$$

where σ_s is the sigmoid activation function, h_{t-1} is the cell output at the previous time-step, W_i and U_i are weight factors, and b_i is the bias.

The forget gate can be computed with Equation (6) [15]:

$$gate(f_t) = \sigma_s(w_t x_t + u_t h_{t-1} + b_t), \tag{6}$$

where W_t and U_t are weight factors and b_t is the bias.

The output is finally computed with Equation (7) [15]:

$$gate(f_o) = \sigma_s(w_o x_t + u_o h_{t-1} + b_o), \tag{7}$$

where W_o and U_o are weight factors and b_o is the bias.

In this study, an LSTM model and an optimized LSTM (PSO-LSTM) model were used to implement daily GHI forecasting. They were compared with other models for probabilistic predictions. The parameters used for optimization are listed in Section 3.4.

The implemented LSTM model was composed of two LSTM models for day-ahead forecasting. One model was to process historical data; the second model was used to process AROME outputs. The outputs of the two LSTM models were concatenated, before being processed by a classical ANN.

3.2. Particle Swarm Optimization (PSO)

Particle swarm optimization was first proposed by Kennedy and Eberhart [20]. This algorithm simulates the predatory actions of a swarm of animals to find the best solution. A massless swarm of particles is created, with only two parameters: their position and speed. Each particle searches for the optimal solution separately in the search space and records it as the current individual extremum. The position of the extremum is shared with other particles in the whole swarm. If one individual extreme value is the best out of all other extremes, it is recorded as the global optimal solution. The global optimal solution is updated every time a particle finds a better extremum.

All the particles in the swarm adjust their velocity and position according to the current extremum already seen by the individual and the current global optimal solution shared by the whole swarm. The formulas for updating the position and speed of the PSO algorithm are shown in Equations (8) and (9) [20]:

$$X_{i,t} = X_{i,t-1} + V_{i,t}, \tag{8}$$

$$V_{i,t} = I_W \times V_{i,t-1} \times c_1 \times \theta_1 \times (pbest_i - X_{i,t-1}) + c_2 \times \theta_2 \times (gbest_i - X_{i,t-1}), \tag{9}$$

where $X_{i,t}$ is the position of the i-th particle during the t-th iteration, and $V_{i,t}$ is the speed of the i-th particle during the t-th iteration. c_1 and c_2 are called the cognitive (personal) and social (global) coefficients, respectively. The coefficients control the exploitation of the individual extremum found by each particle and the levels of exploration made by the swarm in the entire search space. θ_1 and θ_2 are random data, in the range [0, 1]. $pbest_i$ is the best location of the i-th particles among all iterations. $pbest_i$ is the best global location of all particles. I_W is random data initialized in the range [0, 1]

3.3. XGboost

XGboost is a machine learning algorithm realized by gradient lifting technology, and is the first parallel gradient enhanced tree (GBDT) algorithm. XGboost is based on classification and regression tree (CART) theory [21]. It provides parallel tree boosting and is one of the leading machine learning algorithms for regression, classification, and ranking problems. The XGboost model is built by adding trees iteratively. The predicted values of the i-th sample in the t-th iteration can be expressed as follows [21]:

$$\hat{y}_{i,t} = \hat{y}_{i,t-1} + f_t(X_i), \tag{10}$$

where $f_t(X_i)$ represents the addition needed to improve the model. The tree is added iteratively to minimize the objective function, which can be expressed as [21]:

$$obj^{(t)} = \sum_{i=1}^{n} L(y_i, \hat{y}_{i,t-1} + f_t(X_i)) + \Omega(f_t), \tag{11}$$

where $obj^{(t)}$ is the loss function [21].

$$\Omega(f_t) = \gamma T + \frac{1}{2}\lambda \sum_{j=1}^{T} w_j^2, \tag{12}$$

γ and λ are parameters that represent the model complexity. T is the number of leaves, and w_j is a weight parameter.

3.4. Hybrid Models

In this study, a hybrid model, PSO-XGboost, was implemented in order to obtain point forecasts of the GHI. The PSO algorithm is used to choose the best parameters for the XGboost algorithm. Seven important parameters for the XGboost model were chosen, as listed in Table 2. Those parameters were also used by Yu et al. [10] in order to estimate daily reference evapotranspiration values. The parameter "number of trees" has been added, because it is also an important parameter for XGboost.

Table 2. Parameters used in the optimization of XGboost.

Parameters.	Range	Meaning
Learning rate	[0.0001, 0.1]	Step size at each iteration while moving toward the minimization of a loss function.
Number of trees	[1, 500]	Number of trees in XGboost.
Maximum depth	[1, 500]	Maximum depth of a tree. The higher this value, the more likely the model is to overfit.
Subsample	[0.2, 1]	Subsample ratio of the training instances.
Colsample_by_tree	[0.2, 1]	Subsample ratio of columns when constructing each tree.
Min_child_weight	[0, 1]	Minimum instance weight needed in a child.
Gamma	[0.0001, 0.01]	Minimum loss reduction required to make further partitions on a leaf node of the tree.

The ML models were used, in this case, to solve a regression problem; therefore, we set R^2 to be the main metric of the PSO algorithm. R^2 is a positive-oriented metric; thus, the practical objective function used here was $1-R^2$. Indeed, the more the precision of the results increases, the closer R^2 is to 1, which also represents a minimum in the objective function $1-R^2$. Twenty particles are used for the PSO algorithm in order to limit computation time, and to explore the entire research space. The flow chart of the hybrid model is presented in Figure 4.

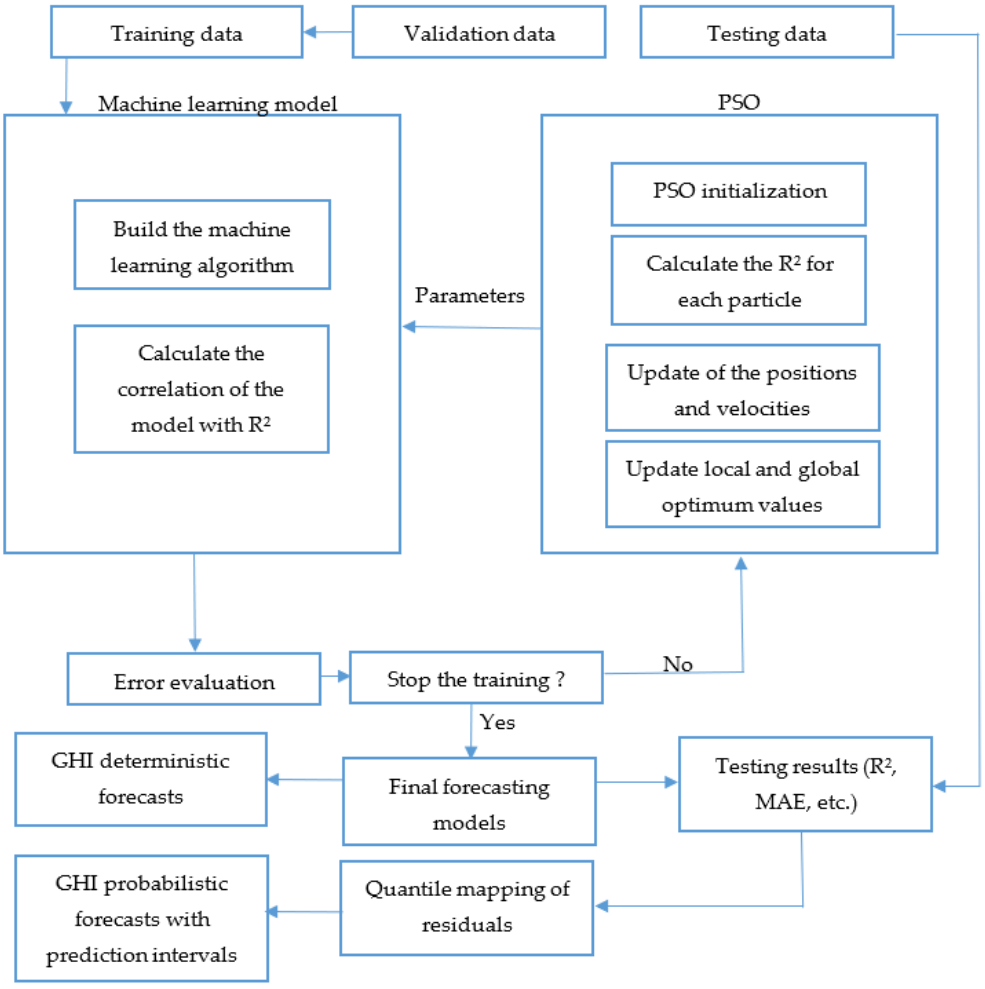

Figure 4. Flow chart of the hybrid models.

A PSO-Gradient boosting model was also implemented, but with fewer parameters than the PSO-XGboost. Only the maximum depth, the learning rate, number of trees, and subsample were used in this instance. The other parameters seen for PSO-XGboost were not available for the gradient boosting algorithm. As stated above, a hybrid PSO-LSTM model was also implemented for daily GHI forecasting. The parameters chosen for the optimization are presented in Table 3.

Table 3. Parameters of the LSTM model for particle swarm optimization.

Parameters	Range	Meaning
LSTM cells in the first LSTM model	[1, 1000]	Number of cells in the first LSTM model.
LSTM cells in the second LSTM model	[1, 1000]	Number of cells in the second LSTM model.
Dropout rate in the first LSTM model	[0.0001, 0.5]	Rate for dropout regularization in the first LSTM model.
Dropout rate in the second LSTM model	[0.0001, 0.5]	Rate for dropout regularization in the second LSTM model.
Dense in the first layer of the ANN	[1, 1000]	Number of neurons in the first layer of the ANN.
Dense in the second layer of the ANN	[1, 1000]	Number of neurons in the second layer of the ANN.
Dropout rate in the ANN	[0.0001, 0.5]	Rate for dropout regularization in the ANN.
Learning rate	$[10 \times 10^{-10}, 10 \times 10^{-2}]$	Step size at each iteration while moving toward a minimum of a loss function.
Epochs	[1, 100]	Number of times the algorithm is trained on the training data.
Validation split	[0.1, 0.5]	Split between training and validation data.

3.5. Residual Modeling

Probabilistic forecasting was implemented in this study through residual modeling. For each individual hour, the residuals were computed and assumed to have either a Gaussian or a Laplacian distribution. This method was inspired by He et al. [22]. The quantiles of the residuals were computed and taken as prediction intervals (PIs). To compute the different quantiles for all the considered distributions, we first needed to consider their cumulative distribution function (cdf) $F_{\text{Residus}}(x)$ in Equation (13).

$$\forall x \in \mathbb{R},\ F_{Residus}(x) = \mathbb{P}(Residus \leq x), \tag{13}$$

The inverse of the cdf is called the percent point function or quantile function $Q(q)$, and is provided in Equation (14):

$$\forall q \in [0,1],\ Q(q) = F^{-1}_{Residus}(x) = inf\{x \in \mathbb{R}, F_{Residus}(x) \geq\ q\ \}, \tag{14}$$

where $Q(0.25)$, $Q(0.5)$, and $Q(0.75)$ are the first quantile, the median, and the third quantile, respectively. The specific quantile function corresponded to a specific distribution (Gaussian or Laplacian). The PIs were calculated at different confidence levels or CLs. In this study, the 38%, 68%, 95%, and 99% PIs were derived from this inverse cdf for the Gaussian distribution in Equation (15). For the Laplacian distribution, the PIs could be derived using Equation (16), defined in [22]:

$$P_{cl+\frac{1-cl}{2}} = \sigma_t Q(cl), \tag{15}$$

$$P_{cl+\frac{1-cl}{2}} = -\sigma_t ln(2(1-cl)), \tag{16}$$

where σ_t is the standard deviation of the distribution (Laplacian or Gaussian). Given the symmetry of those distributions, the upper bounds, U_t, and lower bounds, L_t, were derived using Equations (17)–(19) [22]:

$$U_t = P_{cl+\frac{1-cl}{2}}\ , \tag{17}$$

$$L_t = P_{\frac{1-cl}{2}}, \tag{18}$$

$$L_t = U_t, \tag{19}$$

3.6. Metrics for Probabilistic Forecasting

The quality of probabilistic forecasts was quantified using three different metrics, namely, the prediction interval coverage percentage (PICP), the prediction interval normalized average width (PINAW), and the coverage width-based criterion (CWC), as defined in [13].

The PICP, detailed in Equations (20) and (21), indicates how many real values lie within the bounds of the prediction interval:

$$\text{PICP} = \frac{1}{N}\sum_{i=1}^{N}\delta_i, \tag{20}$$

$$\delta_i = \begin{cases} 1 \ \ if\ y_i \in \ [L_i, U_i] \\ 0\ if\ y_i \notin \ [L_i, U_i] \end{cases}, \tag{21}$$

The PINAW, shown in Equation (22), quantitatively measures the width of the different PIs:

$$\text{PINAW} = \frac{1}{NR}\sum_{i=1}^{N}(U_i - L_i) \tag{22}$$

where R is a normalizing factor. The PINAW represents the quantitative width of the PIs; thus, a lower value of PINAW represents better performance for the prediction intervals.

The CWC, shown in Equations (23) and (24), combines the PICP and PINAW to optimally balance the probability and coverage.

$$\text{CWC} = \ \text{PINAW}\Big(1 + \ \gamma(\text{PICP})e^{-\rho(PICP-\mu)}\Big), \tag{23}$$

$$\gamma(PICP) = \begin{cases} 0 \ \ if\ PICP \geq \ \mu \\ 1\ if\ PICP < \mu \end{cases}, \tag{24}$$

where μ is the preassigned PICP which is to be satisfied, and ρ is a penalizing term. When the preassigned PICP is not satisfied, the CWC increases exponentially. The CWC is a negatively oriented metric, meaning the lower the value, the better.

4. Results

4.1. Preliminary Results

Firstly, before implementing any hybrid model, it is necessary to quantify whether the AROME predictions are effective in increasing the accuracy of our forecasting models. Secondly, a study was also performed to determine how many days should be input into the models, so that we have optimal precision in daily forecasts. These two preliminary results are shown in Table 4 and were only performed for the XGboost model and for lagged terms, from 1 day prior to 5 days prior. The employed metrics were MAE, RMSE, and R^2.

Table 4. Results for lagged days and NWP with the XGboost model.

XGboost Model	Number of Days Prior	MAE (W/m^2)	RMSE (W/m^2)	R^2
Without AROME	1 day	135.19	206.02	0.73
	2 days	136.34	212.47	0.67
	5 days	115.22	182.97	0.74
With AROME	1 day	121.21	194.50	0.76
	2 days	126.29	198.2	0.72
	5 days	**110.24**	**176.47**	**0.76**

The results show that the use of AROME does increase the prediction accuracy for daily GHI forecasts. Indeed, for the same number of lagged days, the results with AROME are always better than without AROME, in terms of MAE, R^2, and RMSE.

With the AROME data, the best values in terms of MAE and RMSE are 110.24 W/m^2 and 176.47 W/m^2 and R^2 = 0.76, respectively, for 5 days prior. For this reason, 5 days prior was taken as a standard way to implement our GHI forecasting tools.

However, it would be interesting to carry out the same study with more lagged days at inputs of the machine learning algorithms. In order to carry out such studies, more historical data and AROME outputs are needed.

The results show a decrease in accuracy for 2 days prior. One possible explanation for this decrease is that the default parameters of the XGboost algorithm might be not ideal for daily GHI predictions for 2 days prior.

4.2. Hybrid Models Results

Tables 5 and 6 present the parameters found by PSO for the XGboost and LSTM algorithms, respectively. Once the optimal parameters are found, the optimized models are tested on the testing data.

Table 5. Optimal parameters for XGboost.

Parameters	Value
Learning rate	0.1
Number of trees	400
Maximum depth	400
Subsample	0.71
Colsample_by_tree	0.99
Min_child_weight	0.96
Gamma	0.01

Table 6. Optimal parameters for LSTM.

Parameters	Value
LSTM cells in the first LSTM model	208
LSTM cells in the second LSTM model	5
Dropout rate in the first LSTM model	0.4
Dropout rate in the second LSTM model	0.15
Density in the first layer of the ANN	712
Density in the second layer of the ANN	786
Dropout rate in the ANN	0.72
Learning rate	0.09
Epochs	50
Validation split	0.5

The results for all the models used for daily GHI predictions are summarized in Table 7 with deterministic metrics, and in Table 8 with probabilistic metrics.

Table 7. Deterministic metrics for all implemented models used for daily GHI predictions.

Models	MAE (W/m^2)	RMSE (W/m^2)	R^2
ANN	120.50	179.40	0.75
CNN	125.66	188.77	0.73
LSTM	115.69	184.74	0.74
Random Forest	106.19	166.42	0.79
Gradient Boosting	111.20	174.54	0.77
XGboost	110.24	176.47	0.76
PSO-Gradient Boosting	105.06	167.24	0.79
PSO-LSTM	**99.37**	154.84	**0.82**
PSO-XGboost	105.02	**153.69**	**0.82**

Table 8. Probabilistic metrics for the implemented models.

Models	Predicted Intervals	Gaussian Distribution			Laplacian Distribution		
		PICP (%)	PINAW (%)	CWC (%)	PICP (%)	PINAW (%)	CWC (%)
ANN	38%	39.81	9.50	9.50	30.95	6.78	13.57
	68%	71.03	18.79	18.79	64.48	16.45	32.90
	95%	94.05	32.77	65.54	94.83	35.30	70.60
	99%	99.60	42.62	42.62	99.80	44.57	44.57
CNN	38%	42.06	10.03	10.03	39.88	7.62	7.62
	68%	71.63	19.37	19.37	67.4	17.30	34.60
	95%	94.64	33.89	67.79	94.24	37.06	74.12
	99%	98.81	45.35	90.71	98.21	46.50	93.00
LSTM	38%	47.81	9.48	9.48	39.88	6.93	6.93
	68%	75.40	18.00	18.00	70.04	15.67	15.67
	95%	93.45	31.37	62.74	94.24	33.85	67.69
	99%	98.21	41.97	83.94	98.02	44.03	88.06
Random Forest	38%	48.61	9.68	9.68	43.85	7.06	7.06
	68%	77.58	18.78	18.78	69.44	15.96	15.96
	95%	94.25	32.35	64.70	93.85	33.47	66.95
	99%	98.61	42.28	84.57	98.02	43.41	86.82
Gradient boosting	38%	48.81	9.99	9.99	43.06	7.28	7.28
	68%	74.80	19.0	19.0	70.63	16.62	16.62
	95%	93.85	32.55	65.11	93.45	34.68	69.37
	99%	98.61	42.31	84.62	97.42	43.97	87.95
XGboost	38%	53.17	9.96	9.96	42.86	7.18	7.18
	68%	75.79	18.94	18.94	69.84	16.3	16.30
	95%	92.46	32.13	64.27	93.85	33.98	67.96
	99%	98.61	41.84	83.69	98.41	43.38	86.76

Table 8. *Cont.*

Models	Predicted Intervals	Gaussian Distribution			Laplacian Distribution		
		PICP (%)	PINAW (%)	CWC (%)	PICP (%)	PINAW (%)	CWC (%)
PSO-LSTM	38%	43.06	8.6	8.6	39.28	6.3	6.3
	68%	73.61	16.86	16.86	69.05	14.68	14.68
	95%	94.63	29.57	59.13	94.44	31.74	63.48
	99%	99.00	40.1	40.1	98.61	42.44	84.87
PSO-Gradient Boosting	38%	52.38	9.63	9.63	43.84	6.93	6.92
	68%	75.20	18.41	18.41	72.42	15.81	15.81
	95%	93.25	31.50	63.01	93.05	33.10	66.20
	99%	98.61	41.40	82.81	97.81	43.19	86.38
PSO-XGboost	38%	45.63	8.84	8.84	39.48	6.6	6.6
	68%	74.01	17.13	17.13	69.84	15.31	15.31
	95%	94.44	30.39	60.78	94.84	33.15	66.30
	99%	98.81	40.61	81.22	98.21	43.23	86.47

For the deterministic metrics, we first note that the use of PSO increases the accuracy of standalone models such as LSTM, GBDT, and XGboost. Indeed, there were decreases in MAE and RMSE and an increase in R^2 when considering standalone models with their optimized versions.

The deterministic metrics also show that the hybrid PSO-XGboost method is the best for implementing daily forecasting, in terms of RMSE = 153.69 W/m^2 and R^2 = 0.82. However, the PSO-LSTM model is also strong, but in terms of MAE = 99.37 W/m^2, as well as R^2 = 0.82. Neither of the two models has any significant advantage over the other.

In order to choose the best model, a Taylor diagram was drawn (Figure 5) for all implemented machine learning models. It can be seen that PSO-LSTM was slightly better than all the other models for deterministic predictions, because it was closer to the observation than the other models. The standard deviation was also the same for the observation and PSO-LSTM (red dotted line), meaning that it appropriately represented the variability in solar irradiance.

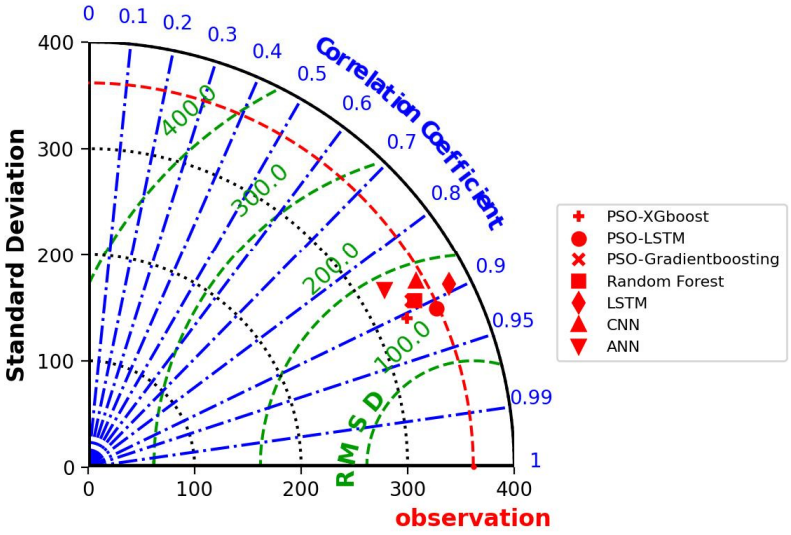

Figure 5. Taylor diagram for deterministic comparison between models.

For all models, we can see that the RMSE is greater than the MAE, which is the manifestation of high variance in individual errors. Indeed, because the RMSE is a quadratic

scoring rule, it tends to assign high weight to large errors, whereas the MAE gives the same weight to all errors, independently of their magnitude. This variance has been studied thanks to residual modelling and the generation of prediction intervals.

For the probabilistic forecasts the PICP, PINAW, and CWC values were computed for all forecasting models. Highlighted in black in Table 8 are the best CWC values, for 38%, 68%, 95%, and 99% PIs. PSO-LSTM is the best algorithm for all prediction intervals. For 38%, 68%, 95%, and 99% PIs, the CWC values are 6.3, 14.68, 59.13, and 40.1, respectively. Notably, for 38% and 68%, the best fit was the Laplacian distribution, whereas for the 95% and 99% PIs, the best fit was the Gaussian distribution. The proposed methods were implemented in Python 3.7, with the machine learning package Tensorflow 2.2.0. Duan et al. [15] also used PSO-XGboost for predicting solar radiation in four different locations in China. After training with four different datasets, the four R^2 values for 1-day-ahead forecasting were 0.816, 0.84, 0.787, and 0.755. Those values are not far from our own PSO-XGboost algorithm, with an R^2 = 0.82. In our case, the PSO-LSTM model was even better than the PSO-XGboost, demonstrating that deep learning models can still outperform ensemble learning models for day-ahead forecasting and, to the best of our knowledge, no PSO-LSTM has ever been used with quantile mapping to obtain day-ahead GHI probabilistic forecasting. The accuracy of point forecasts depends, however, on the global structure of the LSTM, meaning that a simpler structure from an LSTM model might not have the same results.

Figure 6 shows the PSO-LSTM predictions with the corresponding prediction intervals. We can see that the GHI measurements do stay within the prediction intervals; however, we can see that the prediction intervals are quite large. For this problem, it would be interesting to use another method for computing the confidence levels (CLs), which are smaller than the prediction intervals computed in this paper. Li et al. [13] used kernel density estimation (KDE) for confidence level computation, which gave PINAW values of 15.45, 17.03, and 19.55 for 80%, 85%, and 90% CIs, respectively. This is considerably smaller than the PIs in this article, which are approximately equal to 30 for 95% PIs.

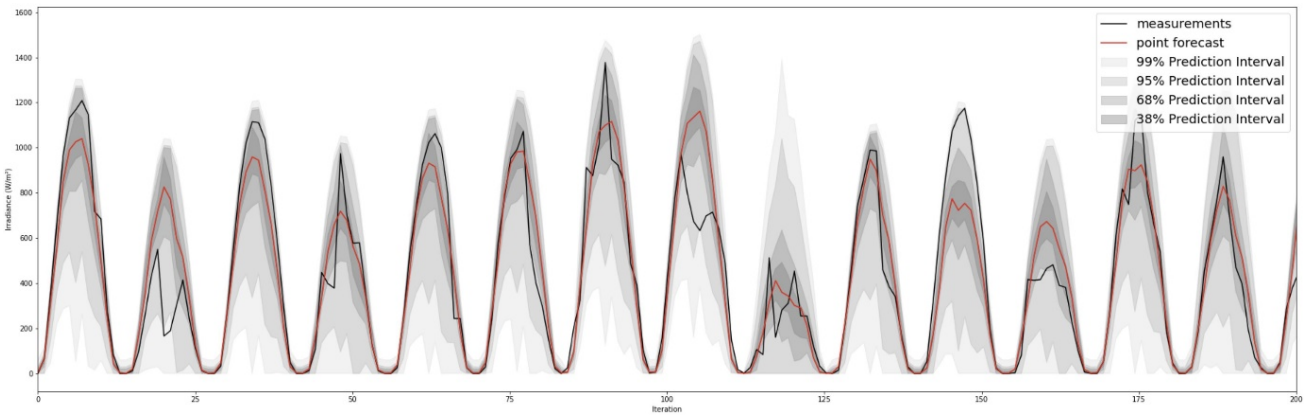

Figure 6. PSO-LSTM probabilistic forecasting.

4.3. Perspectives and Future Research

For the PSO algorithm, the higher the number of particles, the better the exploration of the entire search space; however, the computation time increases accordingly. In order to reduce computation time, we limited ourselves to 20 particles in the swarm. According to Eberhart et al. [23], population sizes ranging from 20 to 50 are optimal in terms of minimalizing the number of evaluations (population size multiplied by the number of iterations) needed to obtain a sufficient solution. Nevertheless, it would be interesting to see the result for GHI day-ahead predictions with the number of particles in a range from 20 to 500 particles for maximum exploration ability of the search space.

As presented in Section 4.1, the maximum precision was obtained for five days of measurements at the input of the models. It is assumed that the more information (lagged days), the better the precision of the forecasting models. For this reason, it would be

interesting to carry out a study with more lagged days fed into the models. However, a constraint arises when more lagged days are used as input vectors. Indeed, with the available data, using more lagged days would greatly reduce the number of training samples. To retain a sufficient number of days for the training of our forecasting models, while simultaneously increasing the number of lagged days, more meteorological data and AROME outputs are needed.

Testing another meta-heuristic model also seems a promising way to improve GHI forecasts. Duan et al. [15] used the Bat algorithm for parameter optimization. Other bio-inspired optimization processes could be implemented, such as the grey wolf optimizer (GWO), whale optimization algorithm (WOA), or salp swarm algorithm (SSA). Duan et al. also showed that the KNEA algorithm is appropriate for providing accurate point forecasts. Therefore, a hybrid model with the metaheuristic models listed above, coupled with the KNEA algorithm, seems to be a very good way to implement daily GHI forecasts. As mentioned in the last section, combining the computation of confidence levels with the KDE method represents a very efficient way of obtaining better probabilistic forecasting from the aforementioned hybrid models.

5. Conclusions

The accurate forecasting of solar irradiance is paramount for photovoltaic power generation. In this study, the solar irradiance forecasts from the operational weather prediction model (AROME), implemented by Météo-France, were compared with in situ measurements for error quantification. In order to drastically improve the forecasting accuracy on-site, to control an isolated solar-powered microgrid called RECIF, implemented in Tahiti, ML algorithms were coupled with a metaheuristic particle swarm optimization (PSO) model for parameter optimization. The novelty of this paper resides in the implementation of probabilistic forecasting by combining an innovative hybrid model (PSO-LSTM) with quantile mapping. Mapping of the residuals allowed us to generate 38%, 68%, 95%, and 99% prediction intervals (PIs) with two different distributions, for probabilistic forecasting. Nine machine learning models were used for comparison purposes, namely, artificial neural network (ANN), convolutional neural network (CNN), long short-term memory (LSTM), random forest (RF), gradient boosting (GBRT), XGboost, PSO-LSTM, PSO-GBRT, and PSO-XGboost. PSO-LSTM was superior to all other models with MAE = 99.37 W/m^2, RMSE = 154.84 W/m^2, and R^2 = 0.82, coupled with a Taylor diagram. The PSO-LSTM model was also the best for all probabilistic metrics, exhibiting a Laplacian distribution for 38% and 68% prediction intervals, with CWC values equal to 6.33 and 14.68, respectively. Furthermore, the PSO-LSTM model showed the best results, exhibiting a Gaussian distribution for 95% and 99% prediction intervals, with CWC values equal to 59.13 and 40.1, respectively. This demonstrates that deep learning models coupled with metaheuristic models can outperform the ensemble learning method for day-ahead GHI forecasting.

Author Contributions: Conceptualization, V.S.; methodology, V.S., D.H. and P.O.; software, V.S. and P.O.; validation, P.O. and D.H.; formal analysis, V.S., P.O. and D.H.; investigation, V.S.; resources, P.O. and M.H.; data curation, V.S. and M.H.; writing—original draft preparation, V.S.; writing—review and editing, V.S., P.O. and D.H.; supervision, P.O. and D.H.; project administration, P.O. and D.H.; funding acquisition, P.O. and D.H. All authors have read and agreed to the published version of the manuscript.

Funding: This study was supported by the EIPHI Graduate School (contract ANR-17-EURE-0002) and the Region Bourgogne Franche-Comté. We thank the National Agency of Research (ANR-18-CE05-0043) for purchasing the equipment needed for this investigation. We also thank the FEMTO-ST laboratory and the University of French Polynesia, for funding this research.

Data Availability Statement: Data are provided within this article.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. IEA. *Electricity Market Report—July 2022*; IEA: Paris, France, 2022. Available online: https://www.iea.org/reports/electricity-market-report-july-2022 (accessed on 2 September 2022).
2. IEA. *World Energy Outlook 2017—Excutive Summary*; Technical Report; IEA: Paris, France, 2017. Available online: https://www.iea.org/publications/freepublications/publication/WEO_2017_Executive_Summary_English_version.pdf (accessed on 3 September 2022).
3. IEA. *Renewable Energy Market Update—May 2022*; IEA: Paris, France, 2022.
4. Ramsami, P.; Oree, V. A hybrid method for forecasting the energy output of photovoltaic systems. *Energy Convers. Manag.* **2015**, *95*, 406–413. [CrossRef]
5. Ehara, T. Overcoming PV Grid Issues in the Urban Areas. 2009. Available online: https://iea-pvps.org/wp-content/uploads/2020/01/rep10_06.pdf (accessed on 2 January 2022).
6. Akhter, M.N.; Mekhilef, S.; Mokhlis, H.; Shah, N.M. Review on forecasting of photovoltaic power generation based on machine learning and metaheuristic techniques. *IET Renew. Power Gener.* **2019**, *13*, 1009–1023. [CrossRef]
7. Ogliari, E.; Dolara, A.; Manzolini, G.; Leva, S. Physical and hybrid methods comparison for the day ahead PV output power forecast. *Renew. Energy* **2017**, *113*, 11–21. [CrossRef]
8. Crisosto, C.; Hofmann, M.; Mubarak, R.; Seckmeyer, G. One-Hour Prediction of the Global Solar Irradiance from All-Sky Images Using Artificial Neural Networks. *Energies* **2018**, *11*, 2906. [CrossRef]
9. Yu, Y.; Cao, J.; Zhu, J. An LSTM short-term solar irradiance forecasting under complicated weather conditions. *IEEE Access* **2019**, *7*, 145651–145666. [CrossRef]
10. Yu, J.; Zheng, W.; Xu, L.; Zhangzhong, L.; Zhang, G.; Shan, F. A PSO-XGBoost Model for Estimating Daily Reference Evapotranspiration in the Solar Greenhouse. *Intell. Autom. Soft Comput.* **2020**, *26*, 989–1003. [CrossRef]
11. Zhang, C.; Ma, Y. (Eds.) *Ensemble Machine Learning*; Springer: Boston, MA, USA, 2012. [CrossRef]
12. Huang, L.; Kang, J.; Wan, M.; Fang, L.; Zhang, C.; Zeng, Z. Solar Radiation Prediction Using Different Machine Learning Algorithms and Implications for Extreme Climate Events. *Front. Earth Sci.* **2021**, *9*, 596860. [CrossRef]
13. Li, X.; Ma, L.; Chen, P.; Xu, H.; Xing, Q.; Yan, J.; Lu, S.; Fan, H.; Yang, L.; Cheng, Y. Probabilistic solar irradiance forecasting based on XGBoost. *Energy Rep.* **2022**, *8*, 1087–1095. [CrossRef]
14. Jia, Y.; Wang, H.; Li, P.; Su, Y.; Wang, F.; Huo, S. Particle swarm optimization algorithm with Gaussian exponential model to predict daily and monthly global solar radiation in Northeast China. *Environ. Sci. Pollut. Res.* **2022**. [CrossRef]
15. Duan, G.; Wu, L.; Liu, F.; Wang, Y.; Wu, S. Improvement in Solar-Radiation Forecasting Based on Evolutionary KNEA Method and Numerical Weather Prediction. *Sustainability* **2022**, *14*, 6824. [CrossRef]
16. Badosa, J.; Wood, J.; Blanc, P.; Long, C.N.; Vuilleumier, L.; Demengel, D.; Haeffelin, M. Solar irradiances measured using SPN1 radiometers: Uncertainties and clues for development. *Atmos. Meas. Tech.* **2014**, *7*, 4267–4283. [CrossRef]
17. Ghimire, S.; Deo, R.C.; Raj, N.; Mi, J. Deep solar radiation forecasting with convolutional neural network and long short-term memory network algorithms. *Appl. Energy* **2019**, *253*, 113541. [CrossRef]
18. Mutavhatsindi, T.; Sigauke, C.; Mbuvha, R. Forecasting Hourly Global Horizontal Solar Irradiance in South Africa Using Machine Learning Models. *IEEE Access* **2020**, *8*, 198872–198885. [CrossRef]
19. Hochreiter, S.; Bengio, Y.; Frasconi, P.; Schmidhuber, J. Gradient Flow in Recurrent Nets: The Difficulty of Learning LongTerm Dependencies. In *A Field Guide to Dynamical Recurrent Networks*; IEEE: Manhattan, NY, USA, 2001; pp. 237–243. [CrossRef]
20. Kennedy, J.; Eberhart, R. Particle swarm optimization. In Proceedings of the ICNN'95—International Conference on Neural Networks, Perth, WA, Australia, 27 November–1 December 1995; Volume 4, pp. 1942–1948. [CrossRef]
21. Chen, T.; Guestrin, C. XGBoost: A scalable tree boosting system. In Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, 13–17 August 2016; pp. 785–794. [CrossRef]
22. He, H.; Lu, N.; Jie, Y.; Chen, B.; Jiao, R. Probabilistic solar irradiance forecasting via a deep learning-based hybrid approach. *IEEJ Trans. Electr. Electron. Eng.* **2020**, *15*, 1604–1612. [CrossRef]
23. Eberhart; Shi, Y. Particle swarm optimization: Developments, applications and resources. In Proceedings of the 2001 Congress on Evolutionary Computation (IEEE Cat. No.01TH8546), Seoul, Republic of Korea, 27–30 May 2001; Volume 1, pp. 81–86. [CrossRef]