energies

Article

A Comparative Study of Time Series Forecasting of Solar Energy Based on Irradiance Classification

Jayesh Thaker [1,*] and Robert Höller [2]

[1] Department of Physics, Carl von Ossietzky Universität Oldenburg, 26129 Oldenburg, Germany
[2] School of Engineering, University of Applied Sciences Upper Austria, 4600 Wels, Austria; robert.hoeller@fh-wels.at
* Correspondence: jayesh.thaker@uni-oldenburg.de; Tel.: +49-15168-797550

Abstract: Sustainable energy systems rely on energy yield from renewable resources such as solar radiation and wind, which are typically not on-demand and need to be stored or immediately consumed. Solar irradiance is a highly stochastic phenomenon depending on fluctuating atmospheric conditions, in particular clouds and aerosols. The complexity of weather conditions in terms of many variable parameters and their inherent unpredictability limit the performance and accuracy of solar power forecasting models. As renewable power penetration in electricity grids increases due to the rapid increase in the installation of photovoltaics (PV) systems, the resulting challenges are amplified. A regional PV power prediction system is presented and evaluated by providing forecasts up to 72 h ahead with an hourly time resolution. The proposed approach is based on a local radiation forecast model developed by Blue Sky. In this paper, we propose a novel method of deriving forecast equations by using an irradiance classification approach to cluster the dataset. A separate equation is derived using the GEKKO optimization tool, and an algorithm is assigned for each cluster. Several other linear regressions, time series and machine learning (ML) models are applied and compared. A feature selection process is used to select the most important weather parameters for solar power generation. Finally, considering the prediction errors in each cluster, a weighted average and an average ensemble model are also developed. The focus of this paper is the comparison of the capability and performance of statistical and ML methods for producing a reliable hourly day-ahead forecast of PV power by applying different skill scores. The proposed models are evaluated, results are compared for different models and the probabilistic time series forecast is presented. Results show that the irradiance classification approach reduces the forecasting error by a considerable margin, and the proposed GEKKO optimized model outperforms other machine learning and ensemble models. These findings also emphasize the potential of ML-based methods, which perform better in low-power and high-cloud conditions, as well as the need to build an ensemble or hybrid model based on different ML algorithms to achieve improved projections.

Keywords: PV power forecasting; probabilistic forecast; machine learning; ensemble models; solar; weather classification; clear sky index

check for updates

Citation: Thaker, J.; Höller, R. A Comparative Study of Time Series Forecasting of Solar Energy Based on Irradiance Classification. *Energies* **2022**, *15*, 2837. https://doi.org/10.3390/en15082837

Academic Editor: John Boland

Received: 22 March 2022
Accepted: 11 April 2022
Published: 13 April 2022

1. Introduction

The output of PV systems is highly variable, especially due to the stochastic formation and movement of clouds in the sky. Energy storage and generation forecasts are the two most important and widely adapted strategies to mitigate the adverse impact of highly variable renewable energy production for the stability of an electricity grid [1]. Accurate PV power forecasting has become a critical factor not only in the overall effective management of electricity grids; it also provides financial benefits in energy market trading, supporting effective decision making based on forecast data.

Accurate and optimized forecasting models based on the use of on-site measurements and data from weather forecast providers are needed [2].

The output of PV power plants mainly depends on solar irradiance (W/m^2) falling on the PV panel. Therefore, global horizontal irradiance (GHI) is the most important input parameter in most PV power prediction systems, and increasing effort is currently spent on research on forecasting of GHI as a basis for PV power forecasts. Different input data and forecasting models are used depending on the forecast horizon.

Numerical methods are based on the actual atmospheric conditions. The dynamic process of weather evolution in a given period is estimated using equations based on fluid thermodynamics [3]. Numerical weather data and satellite images are typically used as input variants for this method. The Global Forecast System (GFS), the European Centre for Medium-Range Forecasts (ECMWF), the North American Mesoscale Model (NAM) and Weather Research Forecast (WRF) are all examples of numerical weather prediction (NWP) approaches [4]. They can be classified as global (worldwide) or mesoscale (local) models based on the area of simulated atmosphere [5].

At a 6 h refresh rate and 1–3 h granularity, these models often yield forecasts for multiple variables important for predicting solar radiation, with forecast horizons ranging from 24 to 72 h or longer. Spatial resolution of these models is variable, and on average, each forecast grid cell is on a scale of a few to hundreds of square kilometers. The quality of NWP forecast is limited by the quality and horizon of the model utilized. Proprietary models, such as satellite-based models or total sky imager models based on cloud cover and cloud motion vectors, are also used to further enhance forecast accuracy. The latest detailed study of solar power forecasting, its dependence on meteorology and grid implications is presented by Yang et al. [6].

Current forecasting methods are based on the individual or small groups of time series. The model parameters for each time series are derived independently from previous observations. Different factors, such as autocorrelation structure, trend, seasonality and other explanatory variables, determine the selection of a particular model. Based on the model dynamics, the fitted model is then utilized to predict the time series into the future. These models also have a scope of probabilistic forecasts through simulation or closed-form expressions for the predictive distributions. Box–Jenkins methodology [7] is widely used for this class of forecasting.

However, instead of predicting individual or a limited number of time series, a new form of forecasting challenge has emerged in recent years, involving forecasting a huge number of related time series [8]. PV power forecasting is one such example that involves time series forecasting of GHI, temperature and several other parameters. In such a case, a large amount of historical data on similar, linked time series can be used to make predictions. Using historical data not only allows for fitting of more complex and potentially more accurate models, but it can also attenuate the time and intensive manual feature-engineering and model-selection steps required by the classical techniques [8].

A plethora of forecasting methodologies for PV power systems has recently been developed and published in the literature. One approach for categorizing forecast methodologies is based on the forecast horizon, which is the amount of time between the actual time and the effective period of prediction [9].

The second classification is based on the forecasting method, which can be statistical or time-series-based, physical, hybrid or ensemble. Statistical approaches are based on a set of observations of one or more parameters taken at a successive sequence of times [10].

Regression methods and persistence methods (primarily used as a standard to test other models) are examples of statistically based forecasters. These methods rely primarily on historical data and do not require any information about PV systems or their installation site. Models such as the autoregressive integrated moving average (ARIMA) work well with non-stationary time series [11]. The drawback of statistical regression models or time series ARIMA models is that they are computationally more intensive and require continuous times series historical data.

According to several reviews, artificial neural network (ANN)-based forecasting is one of the most effective methods. ANNs can capture sharp changes in the input–output relationship due to varying environmental conditions [12] compared with statistical or time series models. However, the fundamental disadvantage is that ANNs necessitate a vast amount of data for training. Any erroneous or insufficient initial data can compromise the predictability of the findings and the accuracy with which the model parameters are chosen. ANNs outperform conventional statistical methods in terms of accuracy and adaptability under intermittent meteorological conditions [12]. A hybrid approach has been applied to enhance the forecasting accuracy of ML and ANN methods by using statistical post processing in conjunction with weather clustering assessment for day-ahead PV power forecasting [13]. Recent publications have also shown interest in forecasting accuracy with varying spatial distribution of PV plants. Various ML models have been compared and examined in terms of technical and economic performance for point forecast with spatial distribution [14].

In this work, the evaluation and comparison of various statistical regression models (SK-Learn linear regression, ridge regression, elastic net), time series models (SARIMAX, Facebook Prophet (FBP)) and ML (gradient boosting (GB), extreme gradient boost (XGB), random forest (RF)) are presented, and forecast equations are derived using GEKKO optimization tool based on the irradiance classification approach to cluster the dataset. The purpose of irradiance classification is to improve on the forecast accuracy, as the irradiance distribution pattern randomly depends on the type and height of clouds. The mentioned models are explained in detail in Section 3. The main contributions of this paper are twofold: (1) The forecast-equation-derived method can be adapted to any location and any technology by deriving the regressive coefficients. (2) This work empirically demonstrates that ML models work better in highly variable cloud conditions; in future, with larger datasets, such models can be used with GEKKO to build a better ensemble model.

2. Materials and Methods

2.1. PV System Description

The ground measurement data employed in this analysis were taken from the PV system of the University of Applied Sciences Upper Austria (FHOOE). The university building is in Wels, Austria (latitude 48.17° N, longitude 14.03° E, elevation 317 m). The system is composed of five types of PV technologies: copper indium selenide (CIS), amorphous silicon (a-Si), cadmium telluride (CdTe), polycrystalline silicon (p-Si) and monocrystalline silicon (c-Si). Figure 1 shows images of the rooftop PV system, as well as the solar radiation measurement station.

The output power of the mono-Si PV system, which is used for the modelling analysis in this paper, is described in Table 1.

Table 1. PV system data.

PV Technology: Silicon mono crystalline	Manufacturer: Sharp
Rated Power: 185 Wp	No. of Modules: 12
Maximum Power of the String STC: 2200 Wp	Inverter: SMA Sunnyboy 1.5/2.0/2.5
Azimuth Angle: 0° south direction	PV Panel Tilt Angle (β): 32°

2.2. Forecasting Methodology

The PV power forecast is based on GHI (W/m^2) and temperature (°C) forecasts with hourly resolution and a forecast horizon of 72 h ahead. These meteorological forecasts ("Blue Forecast") are provided by Blue Sky Wetteranalysen [15]. The Blue Forecast is an automated local weather forecasting tool developed based on statistical methods and meteorological knowhow to correct global GFS-NWP models with historical ground measurements from local weather stations. Local characteristics and conditions such as orography, terrain and climatological factors are used in the model, along with other meteorological features, to enhance the accuracy of global NWP.

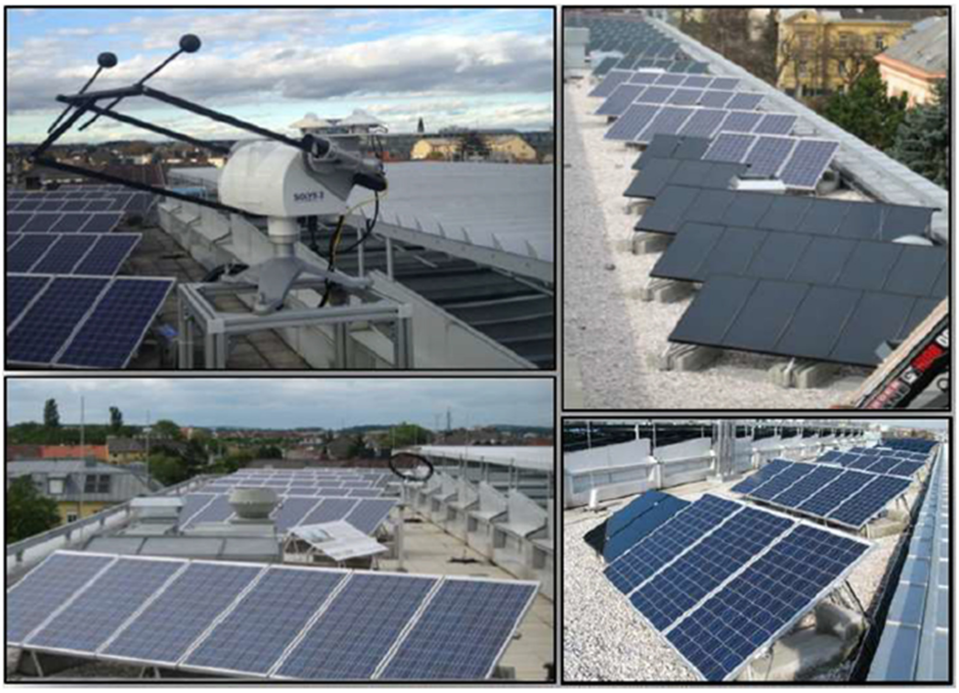


Figure 1. PV system and solar radiation measurement station on the rooftop of the University of Applied Science Upper Austria (FHOOE) at Wels Campus, Austria. Two pyranometers and a pyrheliometer are mounted on a tracker for the measurement of global, diffuse and direct irradiance.

The methodology adopted for this study and the basic steps of the modelling and evaluation algorithm to evaluate power forecast are shown in Figure 2.

The rolling cross-validation approach is used because time series data have a temporal structure, and the data cannot be randomly mixed in a fold while preserving the structure [16].

2.3. Performance Metrics

There are several parameters and metrics used for the evaluation of any solar energy forecasting model to assess the accuracy of the forecasting methods. The three most widely used metrics for forecast accuracy are the mean bias error (MBE), the mean absolute error (MAE) and the root mean square error (RMSE) [17]. The relative MAE (rMAE) and relative RMSE (rRMSE) are calculated by dividing the respective metrics by the average value of the observed parameter, and the normalized metrics (nMAE and nRMSE) are normalized to the maximum variable (power in this case) under consideration.

RMSE is better suited to applications such as grid operation and stability, where high errors are critical [18], whereas MAE is more suited to applications such as day-ahead markets, which have linear costs relative to error [19]. MAE is used as the benchmark metric when calculating the weight factor for a weighted-average ensemble model, as it is appropriate for application with linear cost functions, with the primary goal of day-ahead forecasting with hourly resolution.

The skill score is an important parameter used to evaluate the performance of any model by comparing it with a reference model, which is typically a persistence model in the case of solar forecasting. Mathematically, the skill score (SS) of the forecast is defined by the following equation [20]:

$$SS = 1 - \frac{Score_{forecast\ model}}{Score_{\ Persistence\ model}} \tag{1}$$

where score can be MAE or RMSE.

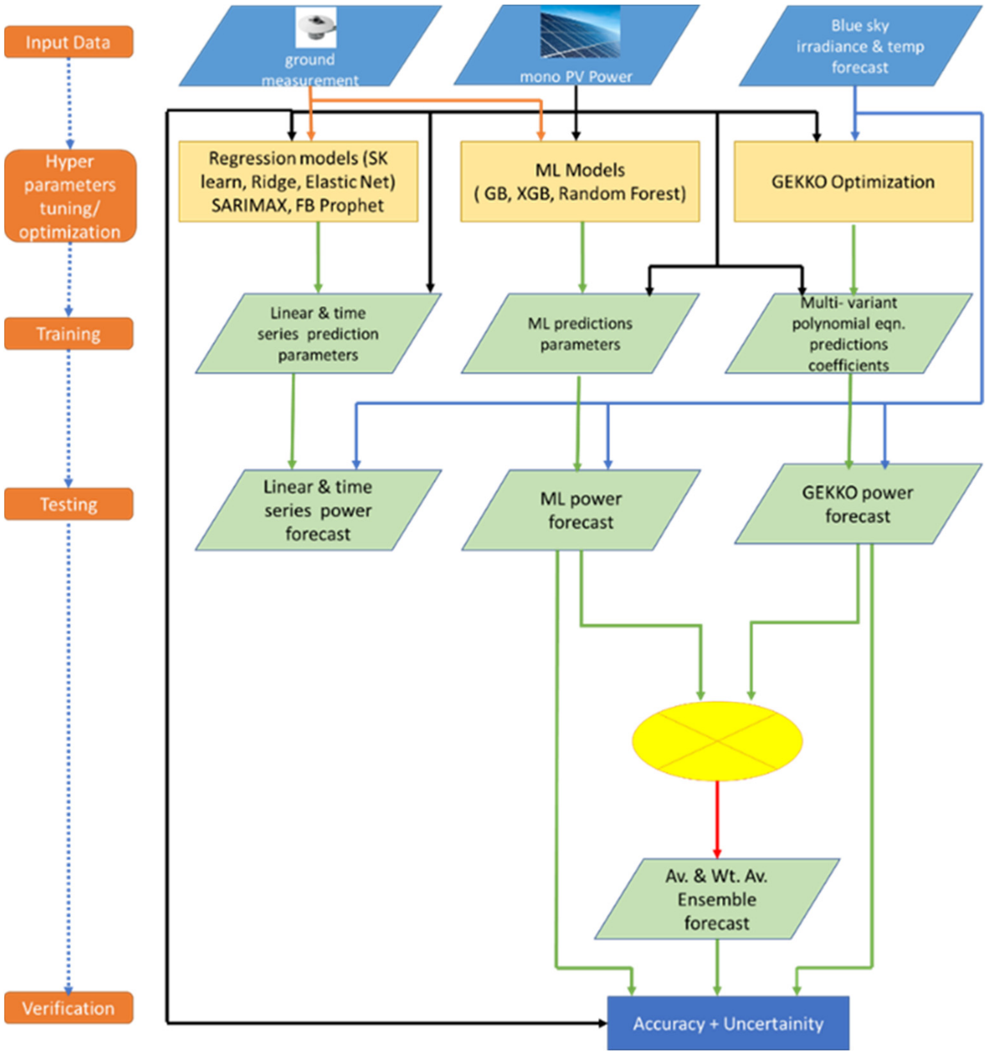

Input Data
Hyper parameters tuning/ optimization
Training
Testing
Verification
ground measurement
mono PV Power
Blue sky irradiance & temp forecast
Regression models (SK learn, Ridge, Elastic Net) SARIMAX, FB Prophet
ML Models (GB, XGB, Random Forest)
GEKKO Optimization
Linear & time series prediction parameters
ML predictions parameters
Multi- variant polynomial eqn. predictions coefficients
Linear & time series power forecast
ML power forecast
GEKKO power forecast
Av. & Wt. Av. Ensemble forecast
Accuracy + Uncertainity

Figure 2. Flowchart of the modelling steps evaluation algorithm.

Specific metrics, such as accuracy, bias, reliability, calibration and sharpness are used for the validation of probabilistic forecasts [21]. Accuracy and bias for point forecasts can also be used for probabilistic forecasts. However, reliability and sharpness are important properties specific to probabilistic forecasts. The similarity between a priori forecast probability and a posteriori observed frequency is measured by reliability. If a forecast appears to be derived from the same distribution as the observation, it is considered reliable. The ability of a forecast to focus the probabilistic information about future outcome(s) is characterized as sharpness. Sharpness is a property of the forecast alone, and its assessment does not involve the observations. Therefore, it is recommended practice that any model's reliability be supported with a sharpness assessment, as otherwise, prediction distribution mis-specifications may go undetected [22].

2.3.1. Probability Integral Transform

Probability integral transform (PIT) is a commonly used metric for reliability. It is performed by converting data values modeled as random variables from any continuous distribution to random variables with a standard uniform distribution [23]. If a random variable (forecast power), X, has a continuous distribution for which the cumulative distribution function is $\hat{F}$, then the random variable, Y, defined as $Y = \hat{F}\ (X)$ has a uniform distribution. Therefore, a uniform PIT histogram means that the forecasting model is perfectly reliable. The shape of the PIT histogram gives information about the reliability index, e.g., a convex U-shaped PIT histogram means overconfident forecasts with an excessively narrow prediction interval, whereas a concave inverse U-shaped PIT histogram means underconfident forecasts with a too widely dispersed forecast distribution and prediction interval [24].

2.3.2. Continuous Ranked Probability Scores

Continuous ranked probability score (CRPS) accounts for the overall accuracy, reliability and sharpness of the forecasts. It is widely used in weather forecasting and atmospheric science, as it allows for comparison of actual results with the whole prediction distribution instead of just comparing to the mean prediction distribution. The standard way to calculate CRPS is given by the integral [25]:

$$CRPS = \frac{1}{N}\int_{-\propto}^{\propto}\sum_{i=1}^{N}\left(\,F_i(x) - \,\hat{F}_i(x)\,\right)^2 \tag{2}$$

where $F_i(x)$ is the cumulative distribution function (CDF) of the probabilistic forecast for the ith value, whereas $\hat{F}_i(x)$ is the CDF of the corresponding ground measurement. Equation (2) has a severe limitation due to the inherent slowness of calculating such an integral for a large database. Gneiting et al. [24] suggested a closed-form solution for a Gaussian distribution, as per the following equation:

$$CRPS\ \left(F_{\mu,\sigma}x\right) = \sigma\left(\left(2\Phi\left(\frac{(x-\mu)}{\sigma}\right)-1\right)+\varphi\left(\frac{(x-\mu)}{\sigma}\right)+\frac{1}{\sqrt{\pi}}\right) \tag{3}$$

2.3.3. Brier Score

The Brier score (BS) is another extensively used skill score that allows for verification and comparison of probabilistic forecasts of dichotomous events. It is defined by the following equation:

$$BS = \sum_{t=1}^{N}(\,p_t - o_t)^2 \tag{4}$$

where p_t and o_t are the respective probabilities of predicted power and observed power that would exceed a fixed threshold. *BS* is also negatively oriented, and *BS* = 0 implies a perfect forecast.

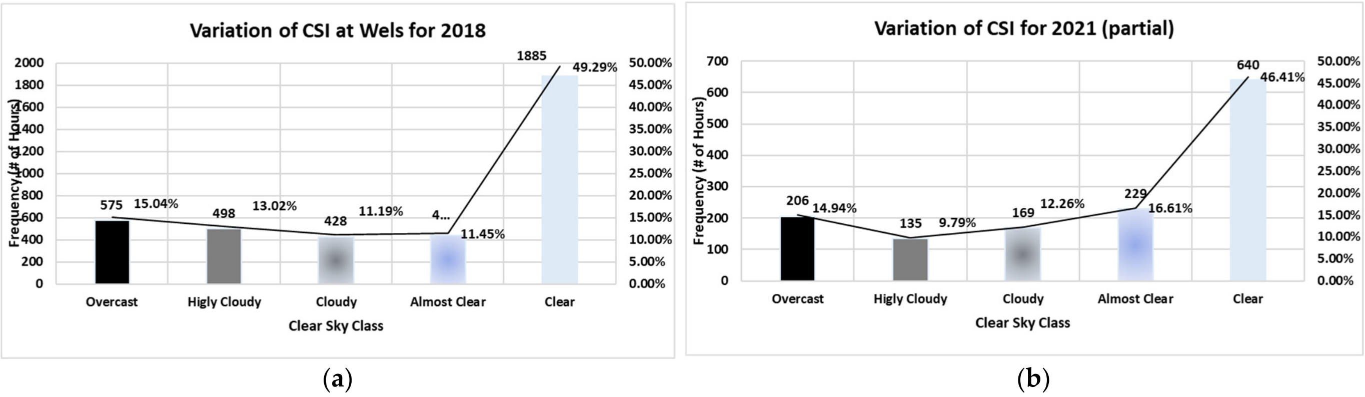
2.4. Database Clustering

The collected GHI and on-site power data measurements are of 10 min resolution and include night hours. For forecasting purposes, the data are averaged to hourly resolution. The PVLIB irradiance clear sky index is used to calculate the clear sky index [26]. The clear sky index is the ratio of global to clear sky global irradiance.

$$K_t = \frac{GHI}{GHI_{csky}} \tag{5}$$

Negative and non-finite clear sky index values are truncated to zero. A maximum value of the clear sky index of 2.0 is applied to account for over-irradiance events typically seen in sub-hourly data due to cloud enhancement and refraction phenomena [26].

We adopted a novel method of classifying the clear sky index into five clusters depending on the value of clearness index, K_t, based on the observation of hourly distribution, K_t, from annual data of Wels for 2018 and the current year trend, as shown in Figure 3. The criteria for the classification of different clusters is illustrated in Table 2.



(**a**) (**b**)

Figure 3. Hourly variation in clear sky index (**a**) for 2018 (**b**) and 2021 (partial data).

Table 2. Classification of sky conditions based on clear sky index.

Cluster Class	CSI Range	Variability
Overcast	$csi < 0.3$	low
Highly Cloudy	$0.3 \leq csi < 0.5$	high
Cloudy	$0.5 \leq csi < 0.7$	high
Almost Clear	$0.7 \leq csi < 0.9$	low
Clear	$csi \geq 0.9$	low

The forecast meteorological data of GHI and temperature are used from 1 May 2021 onwards on a daily basis for 72 h forecast horizon.

The GHI data collected are duly checked and validated. However, pyranometers are highly sensitive to dust, which can lead to an underestimation of GHI. This could possibly lead to a limitation for the lower boundary data of each bin falling into an adjacent lower bin.

There have been calibration and cable problems at the site due to which gaps in the measured data are noted. The recorded ground measurement data used for training and testing are for the periods listed in Table 3. All the models are tested for 96 days during the period, starting from 1 May 2021 until 30 September 2021 for three-days-ahead forecast, and the results are divided into 24 h, 48 h and 72 h forecast horizons for analysis.

Table 3. Training and testing period.

Date (from)	Date (to)
1 January 2021	10 July 2021
22 July 2021	30 July 2021
20 August 2021	25 August 2021
1 September 2021	3 September 2021
21 September 2021	30 September 2021

As the GHI and PV production is 0 for hours before sunrise and after sunsets, these values are excluded from the testing phase by applying a $\cos(\theta_z) > 0$ filtering of the data, where θ_z is the solar zenith angle. $\cos(\theta_z)$ is used in all models as one of the key variables in training and testing, as it strongly influences the GHI.

Month, day of the month, hour of the day, year and season are added as additional feature values, as they define the seasonality of GHI.

GHI, temperature and $\cos(\theta_z)$ values are scaled, whereas all other variables are employed as feature values (binary 0 or 1) in machine learning and linear regression models for training and testing. For GEKKO optimization to derive the equations, forecast values of GHI, temperature and $\cos(\theta_z)$ are used without scaling.

The Fourier transform series are added as features to handle the multiple seasonality. This converts a time and signal function into a frequency and power function. This provides information about which frequencies make up the signal and how strong they are [27]. Fourier transform aids in seeing through the noise and determining which frequencies are important in the real data.

Multiple Fourier series with different k terms in the form of $\sin(2k\pi)$ and $\cos(2k\pi)$ are added, with k value of 5 for each season of the year, hour of the day and week of the year as additional feature values for training and testing ML and times series models.

3. Models

In this study, various statistical linear regression, time series and machine learning models were analyzed and compared to allow for higher degrees of model complexity. This also helps to identify and to model dependency structures. The decision to use these techniques over the many other ML methods is based on a comparison of the models to find the best fit, as well as the possibility of merging them to form ensemble models to increase forecast accuracy. Although different energy forecast models have already been compared [28], the focus of this paper is on improving forecast accuracy through the formulation of prediction equations under different clusters of the clear sky index. Furthermore, the probabilistic forecast of the three best-performing models is evaluated as a measure of uncertainty, and the resulting quantile forecasts are evaluated by different probabilistic forecasting skill scores. The different models applied in this study are:

3.1. Linear Regression

This is the simplest form of a regression model, where the target value is the linear combination of the features [29].

$$\hat{y}(\beta, x) = \beta_0 + \beta_1 x_1 + \cdots + \beta_n x_n \tag{6}$$

Equation (6) fits a linear model with coefficients $\beta = (\beta_1 \ldots \beta_n)$ to minimize the residual sum of squares between the observed and anticipated targets in the dataset.

The coefficients $\beta = (\beta_1 \ldots \beta_n)$ are determined using least square estimation (LSE), which minimizes the following cost function:

$$C_L(\beta) = \sum_{n=1}^{N} (\hat{y}(n) - y(n))^2 \tag{7}$$

3.2. Ridge Regression

Ridge regression is used to estimate the coefficients of multiple regression models, where independent variables are highly correlated [30]. Placing a penalty on the size of the coefficients overcomes some of the issues with regular least squares. The ridge coefficients minimize a penalized residual sum of squares.

A ridge-regularized cost function has two components, i.e., the error term and the regularized term, as illustrated in Equation (8).

$$C_R^{\lambda}(\beta) = C_L(\beta) + \lambda \sum_{i=1}^{n} (\beta_i)^2 \tag{8}$$

The loss is nothing but an error term (first term on the RHS of Equation (8)), followed by a regularized term, λ is a hyperparameter to tune or a balancing factor and β is the sum of squared coefficients. The penalty term (second term on the RHS of Equation (8)) is controlled by changing λ, where λ = 0 implies using only the error term with any penalty, and high value of λ implies the penalty leading to the reduction in the magnitude of coefficients. It is imperative to choose an optimal λ value.

3.3. Elastic Net

LASSO means 'least absolute shrinkage selector operator' and is a linear model that estimates sparse coefficients. The key difference between the LASSO and ridge regression is that LASSO employs the absolute value of coefficients and sets redundant feature coefficients to zero, making it easier to choose the best features.

Elastic net is a fusion of the ridge and LASSO regression techniques. Elastic net regression uses shrinkage to prevent overfitting by minimizing a modified cost function:

$$C_{ElN}^{\lambda_1,\lambda_2}(\beta) = C_L(\beta) + \lambda_1 \sum_{i=1}^{n} |\beta_i| + \lambda_2 \sum_{i=1}^{n} (\beta_i)^2 \tag{9}$$

The second term on the RHS of Equation (9) represents LASSO penalty (L1), whereas the third term represents the ridge penalty (L2), and λ_1 and λ_2 control shrinkage of L1 and L2, respectively.

In a situation with a larger dataset with 500 or more variables, ridge regression only reduces the magnitude of coefficients; if LASSO is used, then reduces coefficients of redundant features to zero (feature selection), which leads to loss of information. Therefore, it is recommended to use elastic net, as it performs well on large datasets. λ_1 and λ_2 are hyperparameters to be tuned by the user.

3.4. SARIMAX

The seasonal ARIMA (SARIMA) model is a generalized form of the ARIMA model that is used to handle seasonality in data. SARIMA uses seasonal autoregressive (AR), moving average (MA) and differencing elements in the model to explicitly deal with seasonality in data. SARIMA is used to model non-stationary series or data that do not oscillate around the same mean, variance or co-variance. As the GHI data (and hence PV power) are not stationary because they has a trend (changing mean) and seasonality, which means that the covariance function does depend on time. The most common way to identify stationarity (or its absence) is by testing a dataset for stationarity with the Dicky Fuller test [31], which tests for a unit root. Basically, if the p-value of the test is too small (i.e., less than 0.05, achieving 95% confidence), the hypothesis that the data are non-stationary is rejected, and data are assumed to be stationary. The standard method of differencing the dataset is adopted to ensure stationarity and then integrated by the same order in as that used in the case of SARIMAX. This model can identify trends and seasonality, which makes it so important.

SARIMAX is seasonal ARIMA with exogenous regressors, which allows it to incorporate the impacts of external factors into the model via an exogenous regressor term. The general form of the SARIMAX model, $(p, d, q)(P, D, Q)_s$, is mathematically represented by Equation (10) [32], where:

- p: is the order of the AR term.
- d: order of differencing to make the data stationary.
- q: order of the MA term.
- P: order of the seasonal AR term.
- D: order of the seasonal differencing to make the data stationary.
- Q: order of the seasonal MA term.
- S: number of periods in a season.

$$y_t = \beta_0 + \beta_1 X_{1,t} + .. + \beta_k X_{k,t} + \frac{\left(1 - \theta_1 B - \theta_2 B^2 - .. - \theta_q B^q\right)\left(1 - \Theta_1 B^S - \Theta_2 B^{2_s} - .. - \Theta_Q B^{Q_s}\right)}{\left(1 - \varnothing_1 B - \varnothing_2 B^2 - .. - \varnothing_p B^p\right)\left(1 - \Phi_1 B^S - \Phi_2 B^{2_s} - .. - \Phi_P B^{P_s}\right)} Z_t \tag{10}$$

where:

- y_t denotes the value of the series at time t.
- $X_{1,t} \ldots X_{k,t}$ denote observations of the exogenous variables.
- $\beta_0, \beta_1 \ldots \beta_k$ is the coefficient value for the k-th exogenous (explanatory) input variable.

- $\varnothing_1, \varnothing_2 \ldots \varnothing_p$ is the weight of the nonseasonal autoregressive terms.
- $\Phi_1, \Phi_2 \ldots \Phi_P$ is the weight of the seasonal autoregressive terms.
- $\theta_1, \theta_2 \ldots \theta_q$ is the weight of the nonseasonal moving average terms.
- $\Theta_1, \Theta_2 \ldots \Theta_Q$ is the weight of the seasonal moving average terms.
- B^S denotes the backshift operator such that $B^s y_t = y_{t-s}$
- Z_t denotes the white noise terms.

3.5. Facebook Prophet

Facebook Prophet uses a decomposable time series model with three main model components: trend, seasonality and holidays. They are combined in the following Equation [33]:

$$y(t) = g(t) + s(t) + h(t) + \varepsilon_t \tag{11}$$

where:

- $g(t)$: piecewise linear or logistic growth curve for modeling non-periodic changes in time series;
- $s(t)$: periodic changes (e.g., weekly/yearly seasonality);
- $h(t)$: effects of holidays (user-provided) with irregular schedules;
- ε_t: error term, accounting for any unusual changes not accommodated by the model.

Using time as a regressor, FBP tries to fit several linear and non-linear functions of time as components. Instead of directly examining the time-based dependence of each observation inside a time series, FBP addresses the forecasting problem as a curve-fitting exercise.

3.6. Random Forest

Leo Breiman and Adele Cutler introduced and patented random forest (RF), which aggregates the output of numerous decision trees to arrive at a single outcome [34]. However, this concept was first proposed in the literature by Ho [35], who employed a consensus of trees generated in random subspaces of the features.

Bagging is a technique to minimize the variance of an estimated prediction function [36]. Bagging works well for high-variance, low-bias procedures. The same regression trees are fit several times to bootstrap-sampled versions of the training data and average the result in case of regression. For classification problems, a committee of trees vote for the projected class.

RF is a significant refinement and expansion of bagging in which a large number of uncorrelated trees is built and then averaged. RF performs similarly to boosting in many problems, but is easier to train and tune, which makes it widely popular and easy to implement in several applications.

The RF algorithm has three critical hyperparameters that need to be tuned before training, viz., node size, the number of trees and number of feature samples. RF classifiers can be used thereafter to solve for regression or classification problems [37].

Classification in RF is performed using ensemble methodology, consisting of collection of decision trees. The training data are fed with a replacement, called the bootstrap sample, to train various decision trees. The correlation among decision trees is reduced by introducing further randomness through feature bagging, thereby adding more diversity. This dataset contains observations and features that will be chosen at random during the splitting of nodes [37].

The RF system relies on various decision trees. Every decision tree consists of decision nodes, leaf nodes and a root node. Each tree's leaf node represents the final output produced by that particular decision tree. The prediction depends on the type of problem, e.g., for a regression task, the individual decision trees will be averaged, and for a classification task, a majority vote, i.e., the most frequent categorical variable, will yield the predicted class. The out-of-bag (oob) sample is then eventually used for cross-validation and prediction [37]. The functioning of RF is illustrated in Figure 4.

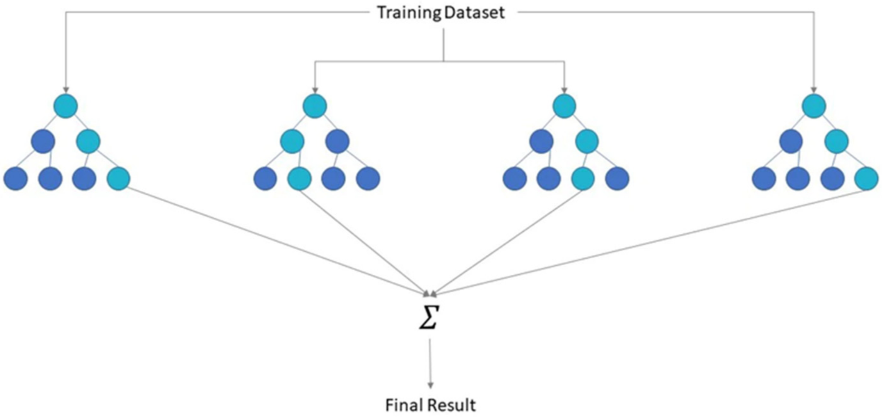


Figure 4. Diagram of random forest classifier (source: IBM).

The essential algorithm for RF applies bagging bootstrap aggregation of the trees. In other words, the primary goal is to average many noisy but approximately unbiased models and hence reduce the variance.

Given a training set, $X = x_1, \ldots, x_n$, with responses $Y = y_1, \ldots, y_n$, bagging repeatedly (N times) selects a random sample with replacement of the training set and fits trees to these samples:

For $n = 1, \ldots ., N$.

1. Sample, with replacement, n training examples from X, Y; (X_n, Y_n)
2. Create a classification or regression tree f_n and train it on X_n, Y_n

After training, the predictions for unknown samples, x', are calculated by averaging the predictions from all the separate regression trees on x' [37]:

$$\hat{f} = \frac{1}{N} \sum_{n=1}^{N} f_b(x') \tag{12}$$

In case of classification, the same task is performed by a majority vote.

The standard deviation of the predictions from all of the individual regression tress on x' is used to measure the forecast uncertainty [38]:

$$\sigma = \sqrt{\frac{\sum_{n=1}^{N} \left(f_b(x') - \hat{f}\right)^2}{N-1}} \tag{13}$$

RF uses a modified tree-learning method to select a random subset of new introduced features at each candidate split in the learning process besides the bagging approach. The reason for doing this is to inject decorrelation among the trees. The study examines how bagging and random subspace projection lead to accuracy gains under various scenarios [39].

Variable m is chosen before each split for a classification in such a way that $m \leq p$ of the input variables at random as candidates for splitting.

Typical values for m are $\sqrt{p}$ or as low as 1. In practice, the best value for these parameters depends on the problem and are usually treated as tuning parameters. The following recommendations are widely used [40]:

- For a classification problem, the default value of m is $\sqrt{p}$, and the smallest node size is 1;
- The default value for m is $p/3$, and the minimum node size is 5 for a regression problem.

3.7. Gradient Boosting

Boosting algorithms were originally introduced for classification problems [40], and subsequently, a statistical viewpoint of boosting was introduced, connecting the algorithm to the loss function [41]. The key approach is to iteratively merge many simple models, known as "weak learners", to produce a "strong learner" with higher forecast accuracy [41]. Boosting was further extended to the regression problem by introducing the gradient boosting machines (GBM) method [42].

Gradient boosting (GB) is a machine learning model that uses the gradient descent optimization method on boosting technique classification, regression and prediction in supervised learning. Boosting differs from RF in that it sequentially constructs an ensemble by training a new weak base learner model (most commonly a decision tree) with respect to the error of the entire ensemble learned so far, as opposed to RF, which depends on simple averaging of models in the ensemble. The error value of a weak classifier is marginally better than that of a random guess. The boosting technique repeats the application of this classifier on modified data, and all predictions are aggregated using a weighted majority vote to obtain a final prediction [43].

In typical supervised learning, given a dataset, $(x, y)_{i=1}^{N}$, where output variable y and vector of input variable x = (x_1, x_2, . . . , x_n), related to each other by some probabilistic distribution function, the objective is to reconstruct functional dependence $x \xrightarrow{f} y$ with an estimate, $\hat{f}(x)$, such that the specified loss function, $\Psi(y, f(x))$, is minimized:

$$\hat{f}(x) = \overset{\mathrm{arg}min}{f(x)} \left(\Psi\left(y, f(x)\right)\right. \tag{14}$$

Once $\hat{f}(x)$ is initialized as constant, the GB algorithm [42] proceeds with the iteration of:

- Computation of so-called pseudo variables or the negative gradient:

$$z_i = -\frac{\partial\Psi(y_i, f(x_i))}{\partial \mathrm{f}(x_i)}|_{f(x_i)=\hat{f}(x_i)} \tag{15}$$

- Fitting a base learner (or a week learner, e.g., tree) regression model, $g(x)$, to pseudo-residuals, i.e., training the model using the training dataset $(x_i, z_i)_{i=1}^{N}$ and predicting z_i from the covariates of x_i;
- Computation of multiplier γ by solving the following one-dimensional problem to choose a gradient descent step size as:

$$\gamma = \overset{\mathrm{arg}min}{\gamma} \sum_{i=1}^{N} \Psi(y_i, \hat{f}_i + \gamma \mathrm{g}(x_i)) \tag{16}$$

- Updating of the model estimate of $f(x)$ as:

$$\hat{f}(x) \ \leftarrow \ \hat{f}(x) + \gamma \mathrm{g}(x) \tag{17}$$

The algorithm determines the direction and the gradient in which it needs to improve at each iteration to fit the data and select a particular model.

The trees are iteratively fitted to the residuals using GB to optimize any arbitrary differentiable loss function, such as the squared error cost function, where the negative gradient is equal to the residuals [44].

The GBM algorithm performs better if at each iterative step, the contribution of the added decision tree is shrunk by using a shrinkage parameter, α, also known as the learning rate, which ranges between 0 and 1. This helps to constrain the fitting procedure and address the overfitting problem [45].

3.8. Extreme Gradient Boosting

Extreme gradient boosting (XGB) is comparable to GBM in that it uses the same gradient boosting technique to construct an ensemble of trees [46]. There are, however, important differences and a few advantages of XGB over GBM [47], such as:

- Regularization: XGB offers additional regularization hyperparameters that provide added protection against overfitting.

- Early stopping: XGB implements early stopping, allowing for stopping of the process of modelling when additional trees offer no improvement.
- Parallel Processing: It is particularly difficult to parallelize GBM because of its sequential nature. XGB has introduced methods to support GPU and Spark compatibility, which enables the use of distributed parallel processing to fit gradient boosting.
- Loss functions: Using specific objective and evaluation criteria, users can define and optimize gradient boosting models in XGB.
- Continue with existing model: A user can train an XGB model, save the results and later return to that model and continue building onto the results, thereby allowing continued training of the model without starting from scratch.
- Different base learners: Most GBM implementations are built with decision trees, but XGB also provides boosted generalized linear models.

The XGB algorithm proceeds with the following iterations once $\hat{f}(x)$ is initialized as constant using Equation (14):

- Computation of gradients and hessians as:

$$g_i = \frac{\partial\Psi(y_i, \mathrm{f}(x_i))}{\partial \mathrm{f}(x_i)}|_{f(x_i)=\hat{f}(x_i)} \tag{18}$$

$$h_i = \frac{\partial^2\Psi(y_i, \mathrm{f}(x_i))}{\partial \mathrm{f}(x_i)^2}|_{f(x_i)=\hat{f}(x_i)} \tag{19}$$

- Fitting a base learner (or a week learner, e.g., tree) regression model, $g(x)$, to pseudo-residuals, i.e., training the model using the training dataset $(x_i - \frac{g_i}{h_i})_{i=1}^{N}$ by solving the following optimization problem:

$$\hat{\varnothing} = \overset{\mathrm{arg}min}{\varnothing} \sum_{i=1}^{N} \frac{1}{2}\hat{h}_i[-\frac{\hat{g}_i}{\hat{h}_i} - \varnothing\,]^2 \tag{20}$$

$$\hat{f}(x) = \alpha\,\hat{\varnothing}\,(x) \tag{21}$$

- Update the model estimate of $f(x)$ as:

$$\hat{f}_{(m)}(x) = \hat{f}_{(m-1)}(x) + \hat{f}_{(m)}(x) \tag{22}$$

- Model output:

$$\hat{f}(x) = \hat{f}_{(M)}(x) = \sum_{m=0}^{M} \hat{f}_m(x) \tag{23}$$

A detailed overview, implementation suggestion and hyperparameter tuning for GBM and XGB are presented by Boehmke [47].

3.9. GEKKO

GEKKO is a mixed-integer and differential algebraic equation optimization software tool. It provides large-scale solvers for linear, quadratic, nonlinear and mixed-integer programming (LP, QP, NLP, MILP, MINLP). Data reconciliation, real-time optimization, dynamic modeling, and nonlinear predictive control are some of the modes of operation [48,49].

GEKKO is a package that includes the ability to run model-predictive control, dynamic parameter estimation, real-time optimization and parameter update for dynamic models in real-time applications. It provides an algebraic modeling language (AML) for solving optimization problems in simple object-oriented, equation-based models to interface with powerful built-in optimization solvers [48].

GEKKO uses nonlinear approaches in model development, dynamic data reconciliation and dynamic optimization to meet research and industrial application objectives [50].

The interface between advanced solvers and human users is generally facilitated by AML. Variable limits, constraint functions and bounds, goal functions, and first and second derivatives of the functions, all in consistent array format, are required by high-end, off-the-shelf gradient-based solvers. AMLs make the process easier by allowing the model to be written in a straightforward, user-friendly manner. AML takes in a model (constraints) and an optimization goal [48].

The GEKKO optimization tool solves a problem in the form Equation (24) using AML design.

$$\begin{array}{c} \underset{u,x}{min} \quad J(x,u) \\ 0 = f(x,u) \\ 0 \ \leq g\ (x,u) \end{array} \tag{24}$$

By tuning and optimizing the state variables, x, and inputs, u, the objective function in Equation (24) is minimized. The inputs, u, typically include variables such as measured disturbances, unmeasured disturbances, control actions, feed-forward values and parameters. The solver determines these parameters to minimize the objective function, J. Differential or algebraic equations, which include equality constraints (f) and inequality constraints (g), are used based on the application to solve the state variables, x [48].

A detailed overview of GEKKO with nine different modes of operation for different solutions and applications is presented in [48].

The basic set of variable types includes constants, parameters and variables (with initial values and bounds). Fixed variables (FV) inherit parameters but have the possibility to add a degree of freedom, and they remain constant over the horizon. The built-in optimizer implicitly solves all user-defined equations simultaneously. Equations are discretized across the entire time horizon in dynamic modes, and all time points are solved at the same time [48].

The following settings are applied in the optimization algorithm in this study to estimate the multivariate polynomial coefficients to derive the equation between PV power output as a function of forecast GHI, forecast temperature, forecast clear-sky index and $\cos(\theta_z)$:

- Variable type: fixed variable;
- Equation: user-defined;
- Optimization simulation type: non-dynamic steady state;
- Estimation: model parameter update $(IMODE = 2)$;
- Measure power value to provide an initial guess: $(y.STATUS = 1)$;
- APOPT solver: $(m.SOLVER = 1)$, as the number of degrees of freedom is less than 2000.

After multiple iterations with various equation formats, the following forecast equation is generated:

$$\begin{array}{c} \hat{y} = ax_1 + bx_1^2 + cx_1^3 + dx_2 + ex_2^2 + fx_2^3 + gx_3 + hx_3^2 + ix_3^3 + jx_4 + kx_4^2 + lx_4^3 \\ \hat{y} = 0\,;\ if\ x_1 = 0 \\ \hat{y} = 0\,;\ if\ x_3 \leq 0 \end{array} \tag{25}$$

where:

$\hat{y}$ = Forecast power;
x_1 = Forecast GHI;
x_2 = Forecast temperature;
x_3 = $\cos(\theta_z)$;
x_4 = Forecast clear-sky index.

Five sets of coefficients of Equation (25) optimized and derived using GEKKO for different sky conditions by applying clustering on forecast clear-sky index as per Table 2 are listed in Table 4.

Table 4. GEKKO-optimized coefficients for different sky conditions.

Coefficients	Clear Sky Conditions				
	Clear	Almost Clear	Cloudy	Highly Cloudy	Overcast
a	1.067684687	−0.461843016	−2.828685511	11.89417827	12.06287803
b	0.00025502	0.007193564	0.009622766	−0.022698139	−0.09306182
c	-1.04×10^{-6}	-5.42×10^{-6}	-5.35×10^{-6}	2.63×10^{-5}	0.000287555
d	−1.19666093	101.6616766	20.19126047	55.41016311	122.701323
e	0.139029555	−6.15386442	−1.77577578	−6.238438444	−10.7498668
f	−0.00652831	0.101729804	0.040920868	0.184678849	0.287472093
g	−890.843258	830.2304169	800.9744453	−2954.010321	−892.164004
h	4878.199184	−1303.08823	1879.956472	1135.067326	435.5724609
i	−2772.24594	985.3253282	−1807.3081	−8.943386893	231.1234833
j	29.6447275	4332.245064	−1373.59256	1948.511962	−6901.24578
k	3.28850648	−13063.3550	3732.719303	−7003.472994	33939.9029
l	−6.25419753	8462.759824	−2893.75276	4896.689361	−58542.1699

3.10. Ensemble Model

As illustrated in Section 4, the performance of linear and time series regression models is much lower than that of ML models and GEKKO; therefore, only RF, RF with Fourier, GB, XGB and GEKKO models are considered for the evaluation of average and weighted averaged ensemble models. In an average ensemble model, the average forecast of each individual model is evaluated under different clusters based on the forecast value of clear sky index, i.e.,

$$\hat{y}_{av} = \frac{\hat{y}_{RF} + \hat{y}_{RFF} + \hat{y}_{GB} + \hat{y}_{XGB} + \hat{y}_{GEK}}{5} \tag{26}$$

In the weighted average ensemble, the weight of each model performance is first evaluated for different clusters by using the inverse value of MAE, e.g., the weight of RF forecast:

$$w_{rf} = \frac{IMAE_{rf}}{IMAE_{rf} + IMAE_{rrf} + IMAE_{gb} + IMAE_{xgb} + IMAE_{gek}} \tag{27}$$

where IMAE = 1/MAE.

The weighted average ensemble forecast is then evaluated on an hourly basis for each cluster based on the forecast value of clear sky index as:

$$\hat{y}_{wav} = \frac{w_{rf}\hat{y}_{RF} + w_{rff}\,\hat{y}_{RFF} + w_{gb}\,\hat{y}_{GB} + w_{xgb}\,\hat{y}_{XGB} + w_{gek}\,\hat{y}_{GEK}}{w_{rf} + w_{rff} + w_{gb} + w_{xgb} + w_{gek}} \tag{28}$$

4. Results and Discussion

4.1. Overview and Analysis

In this section, the evaluation of the power forecasts obtained by the different models described in Section 3 is presented, and a comparison of different methods is performed to assess which models perform best under given conditions. This is followed by a more detailed analysis with respect to the performance of the forecasts for different clear sky index clusters, as well as monthly forecasts. Finally, information on the variation in performance of each model with increased forecast horizon and the probabilistic forecast for the best performing models is provided.

Figure 5 shows the box plot for the aggregate forecast data for different models compared with the actual measured value (y) and Figure 6 shows rMAE and rRMSE and associated skill scores for a 24 h forecast horizon for the entire test period. It is evident that GEKKO outperforms all the models both in rMAE (20.9%) and rRMSE (32.4%), with the skill score of both indices close to 0.48.

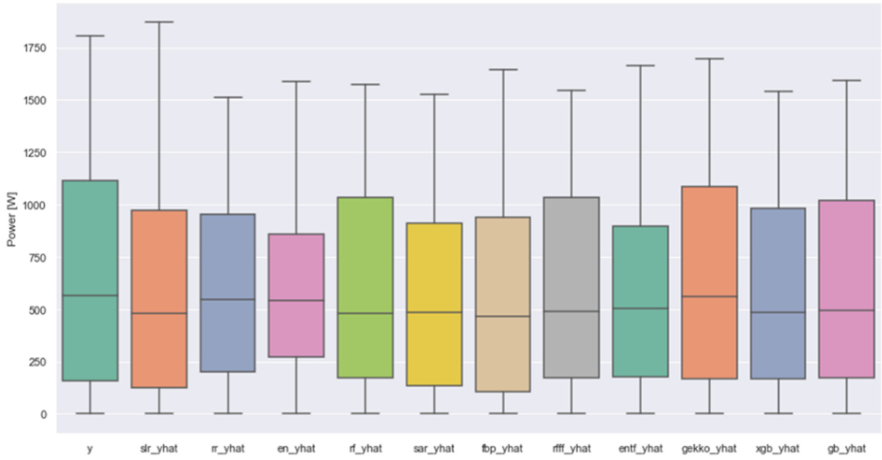


Figure 5. Box plot of aggregate power forecast comparison of different models for a 24 h horizon. (y: measured power; yhat: forecast power; slr: sklearn linear; rr: ridge regression; en: elastic net; rf: random forest, sar: sarimax; fbp: Facebook Prophet; rfff: random forest with Fourier; entf: elastic net with Fourier; xgb: extreme gradient boost; gb: gradient boost).

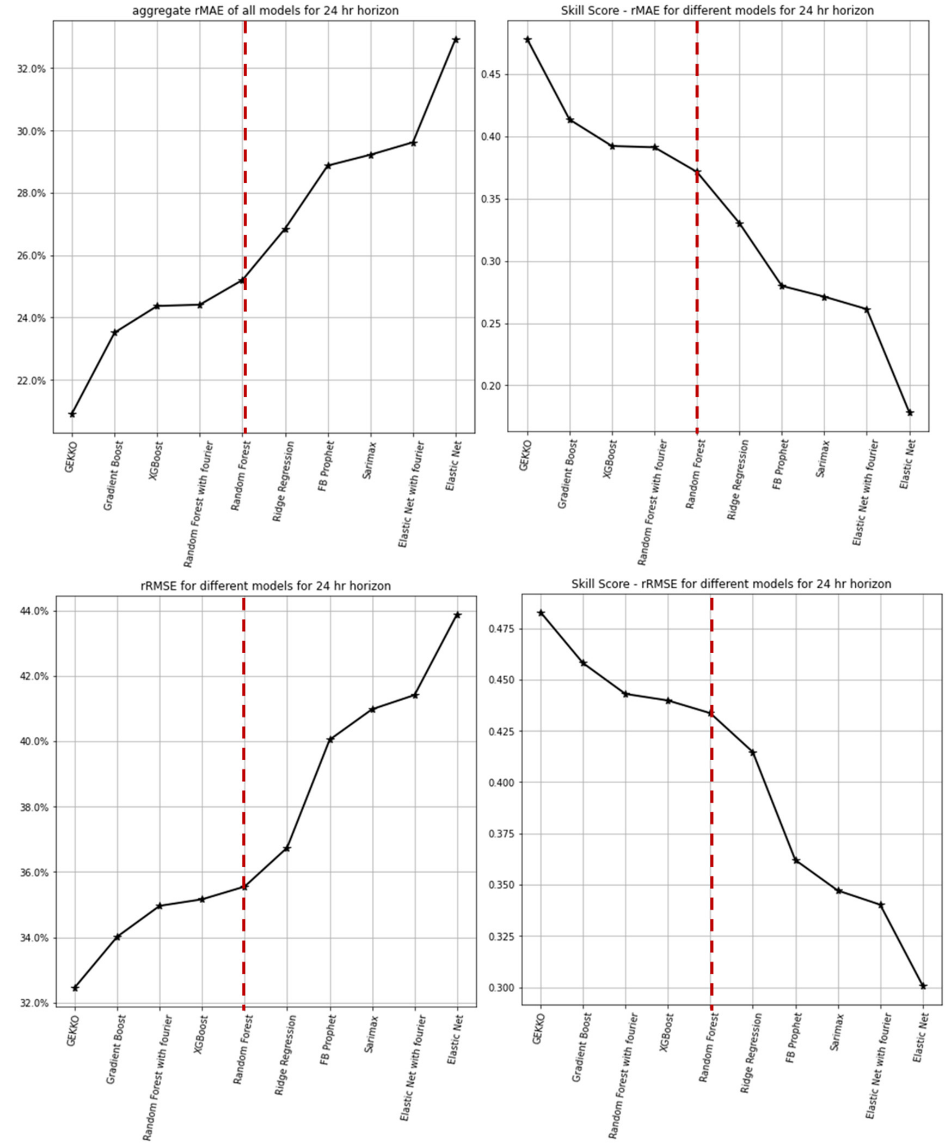


Figure 6. rMAE, rRMSE and corresponding skill scores for all models for a 24 h horizon.

Linear regression models SKlearn, ridge and elastic net and time series models SARIMAX and FBP did not perform well. These models apparently require continuous data without any serious gaps, and as such, their forecast performance deteriorated due to gaps in the measured data, as explained in Section 3.3. Solar power is not a linear function of the GHI, and as such, the performance of the linear models deteriorates compared with ML and polynomial regressions models. Furthermore, as the purpose is to build a robust forecast model with minimal dependence on historical data, the focus lies on further analysis of the five best-performing models, i.e., GEKKO, GB, XGB, RF and RFF.

Figure 7 shows important metrics under different clear sky conditions and the aggregate during the whole test period. Normalized indices are calculated by dividing the respective metrics by installed power capacity, i.e., 2200 W.

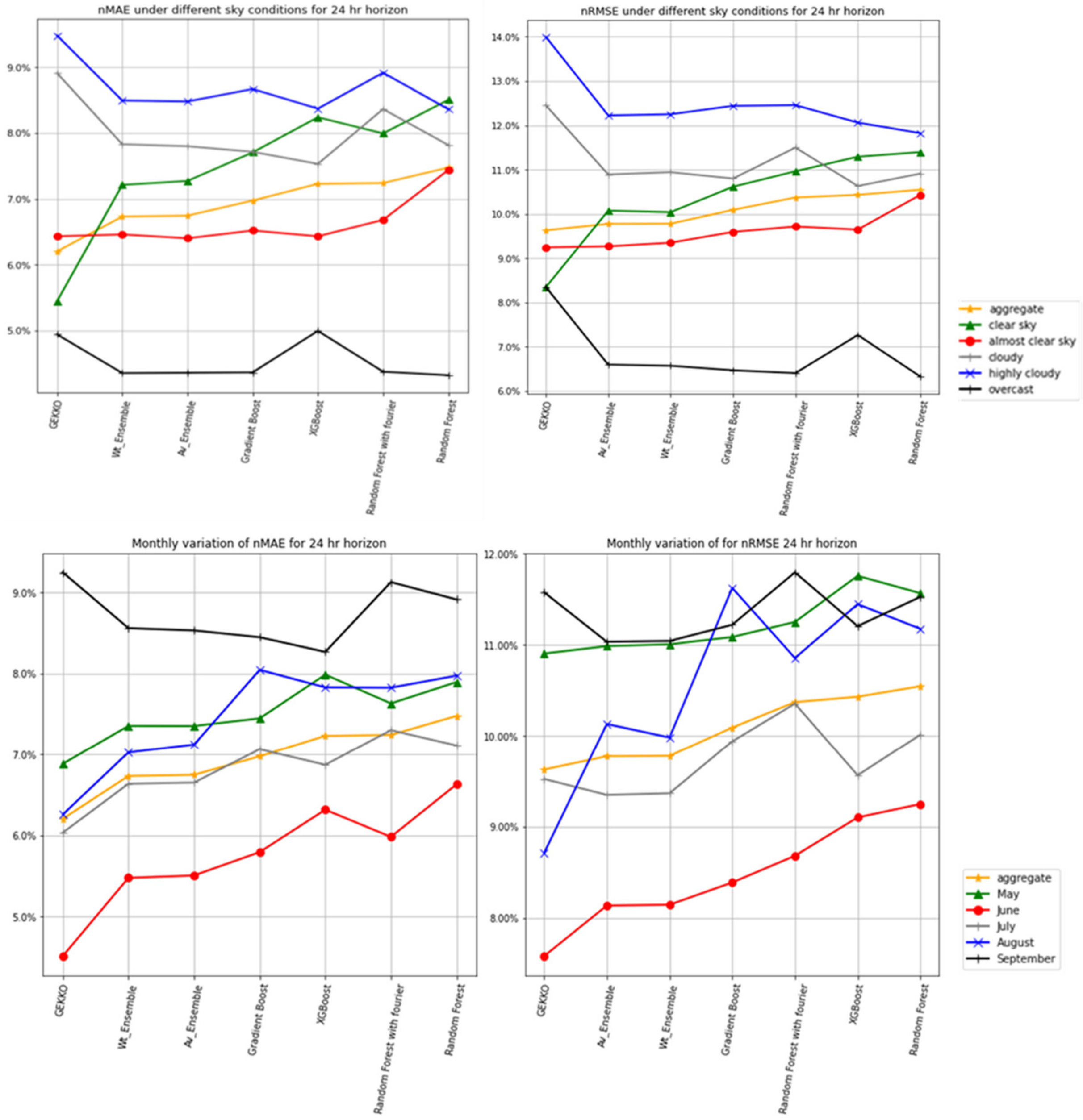


Figure 7. Metrics: aggregate and under different clear sky index clusters and monthly for 24 h horizons.

It is evident that overall (aggregate), GEKKO outperforms all other models for all metrics. Furthermore, GEKKO outperforms other models under clear sky and almost clear sky conditions, i.e., for clear sky index ≥0.7, whereas the weighted average ensemble model performs the best under cloudy, highly cloudy and overcast conditions, i.e., under clear sky index <0.7. The performance of ML models (GB, XGB, RF) is close, with the advantage of introducing multiple Fourier transform series as additional features evident from the fact that RF with Fourier performs better than RF, especially under highly variable cloudy and highly cloudy conditions.

The month-wise analysis shows results in line with expectations that all models perform better than the aggregated model during summer (June and July), as there are more instances of clear sky hours during these months compared to other months (May, Aug, Sept), where the performance of all the models drop compared to the aggregate model.

It is evident from hourly variation analysis of metrics (Figure 8) that GEKKO outperforms all the models from 8:00 to 14:00 h, followed by the weighted and average ensemble models. The performance of GEKKO is relatively poor in the early morning hours from 6:00 to 8:00 h. This is corroborated by the analysis of the distribution of the forecast clear sky index, which shows a higher density of forecast, $K_t < 0.5$, during the period. The statistical difference in the performance of all the models is insignificant from 16:00 to 18:00 h.

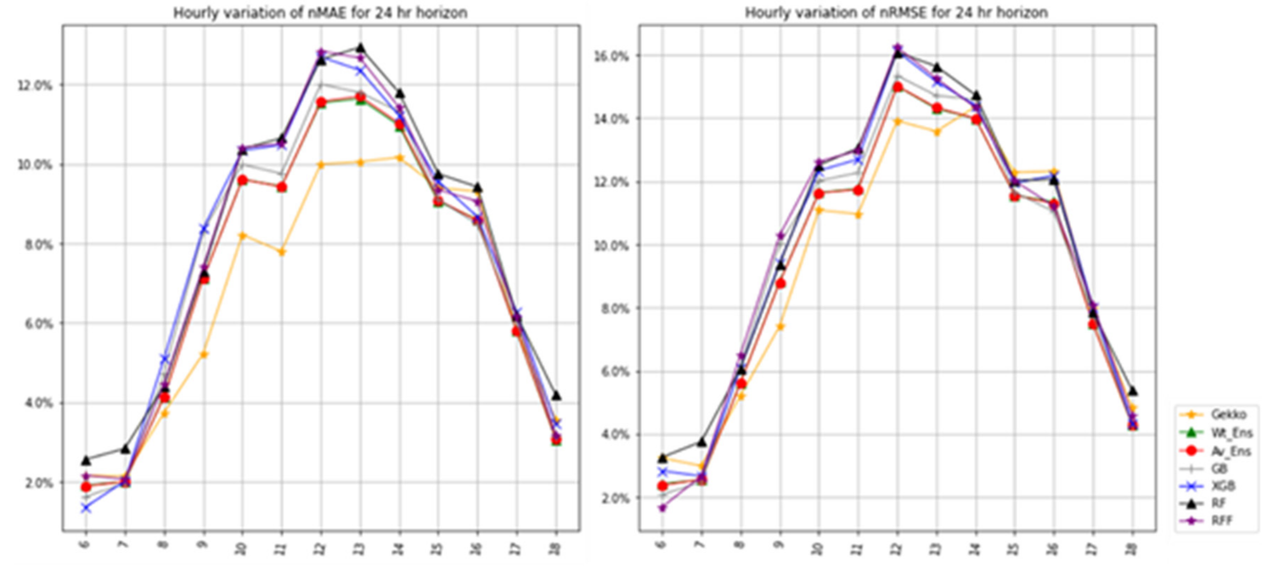


Figure 8. Hourly variation in nMAE and nRMSE of different models for 24 h horizon.

Figure 9 illustrates the variation in aggregate nMAE and nRMSE of different models for 24 h, 48 h and 72 h forecast horizons. As anticipated, the forecasting error rises as the forecast horizon increases for all metrics.

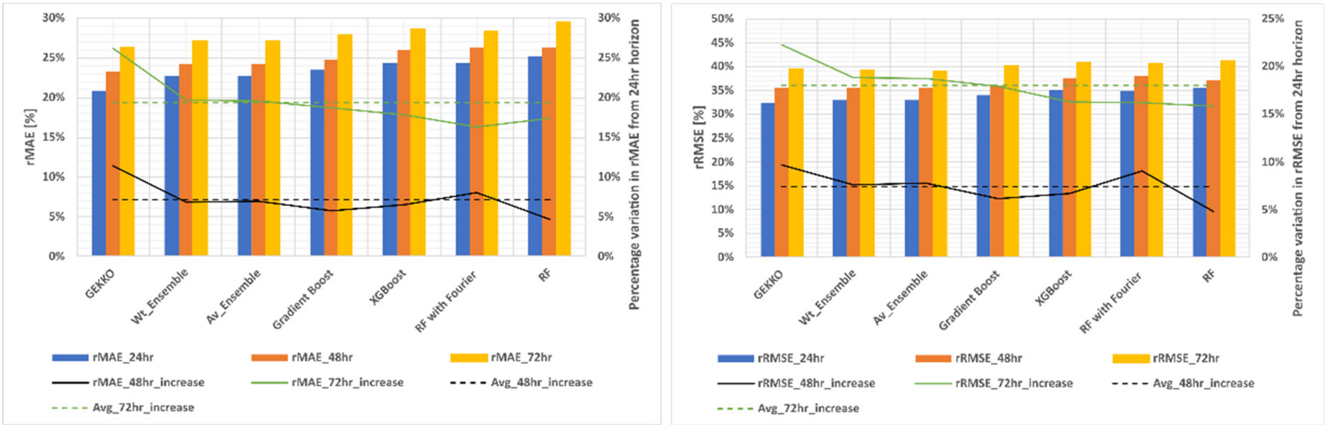


Figure 9. Variation of aggregate nMAE and nRMSE of different models for different forecast horizons.

Detailed analysis of Figure 9 shows that the drop in performance of GEKKO is much higher compared with that of other models. The nMAE for GEKKO increases by 11% and 26%, whereas for all other models, the increase in nMAE is around 7% and 19% for 48 h and 72 h forecast horizons, respectively, compared to a 24 h forecast horizon. A similar observation is made on the other scores. This is because the coefficients in Table 3 derived for GEKKO are based on the forecast data for a 24 h horizon, and the same coefficients and equations are used to forecast for 48 h and 72 h horizons, as the main purpose of this study is to forecast for a 24 h horizon. Nevertheless, GEKKO still performs better than other models for 48 and 72 h horizons.

4.2. Probabilistic Forecast

The forecasting accuracy of PV power greatly depends on the forecasting quality of meteorological parameters, particularly on GHI and temperature. Therefore, uncertainty is bound to play a major role in forecasting of power because of the associated uncertainty in meteorological parameters due to the stochastic nature of weather conditions. Probabilistic forecasting is therefore very important in quantifying the forecast uncertainty by delivering not only the deterministic forecast but also a full predictive distribution.

A probabilistic forecast enables users to adapt their decisions based on additional uncertainty information. For example, the uncertainty in forecast estimates is critical in determining the best bidding approach for a microgrid with a high proportion of renewable energy sources [51]. A study showed that probabilistic forecast for day-ahead energy bidding of an Italian wind farm can result in a 23% increase in the annual profit in comparison to point forecasts [52].

A point forecast with a confidence interval is desirable to show the claimed accuracy [53]. For a reasonably large sample size, an x% confidence interval, by definition, gives the probability that future ground measured data will fall in x% confidence interval for the forecast under simulated conditions, with an assumption that the model is correct. Statistically, it is defined as [53]:

$$Confidence\ interval = forecast\ \pm (critical\ t - value) \times (std.\ forecast\ error) \tag{29}$$

The lower and upper quantile limits for the 95% confidence interval are calculated using a t distribution for GEKKO and weighted ensemble probabilistic forecast comparison and analysis. GB is one of several ML models that can be adapted to predict quantiles. The GB tree-boosting algorithm predicts α quantile with an appropriate quantile cost function and hence can be used for quantile regression [54]. As the primary focus of this work is to compare the probabilistic forecasts of the models, quantile GB is also used to predict two quantiles, i.e., 0.05 and 0.95.

Figure 10 shows the intraday time series plot of forecast with a 95% confidence interval for a 24 h forecast horizon, along with measured power for eight consecutive days starting at 21 September 2021 for the three models under consideration for probabilistic forecast. The light-blue shaded region between the red and orange lines is the 95% confidence interval. The statistical analysis shows that 90.6% of the ground-measured data fall within the 95% confidence interval of GEKKO, whereas the same is 89.8% and 88.3% for weighted average ensemble and GB, respectively. There is also a statistically significant higher percentage of absolute errors in the lower quantile for all three models.

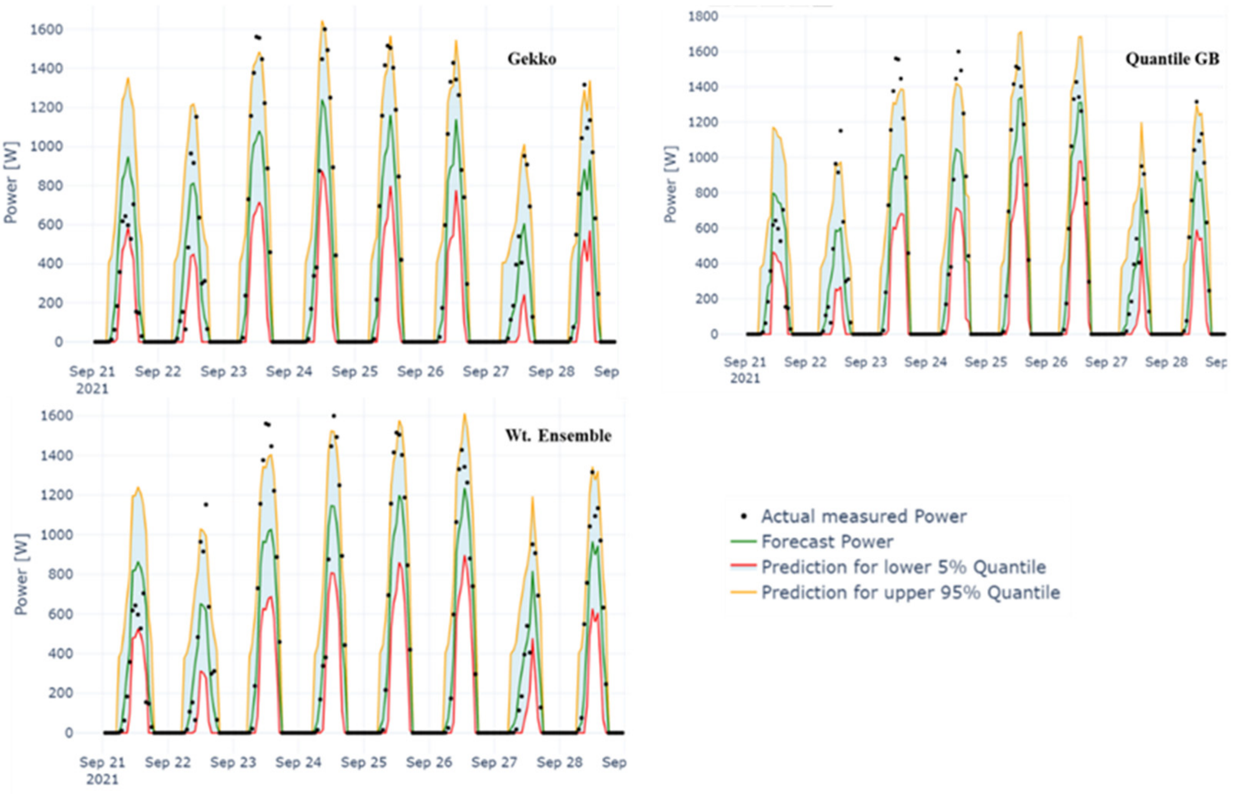


Figure 10. Intraday forecast for 24 h horizon-time-series plot forecast power (green) with 95% confidence interval measured power (black dots) for GEKKO, QGB and weighted ensemble models.

Figure 11 shows the PIT histograms of GEKKO, weighted ensemble and GB models with a 95% confidence interval for each bin shown as a red error bar, with the dashed line representing significant level 1 of uniform distribution. The high value on the first bin for all the models suggests that these models underestimate the lower quantile. The performance of GEKKO is almost 100% uniform in bins 0.5–1.0, suggesting its excellent reliability skill in the respective quantiles.

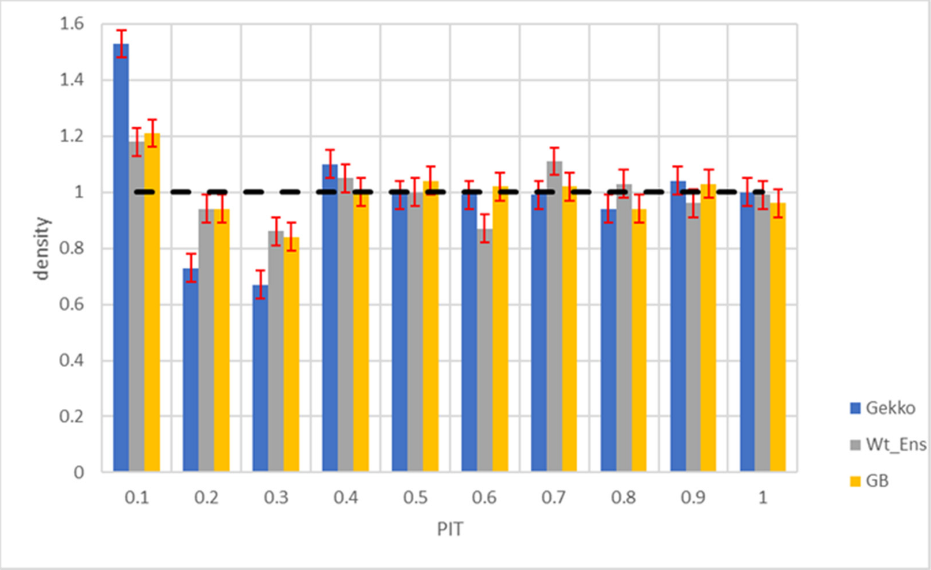


Figure 11. PIT histograms of GEKKO, weighted ensemble and GB models.

Figure 12a shows the relative CRPS (rCRPS) of three models for overall and different sky conditions. rCRPS is calculated by dividing the CRPS score by the average value of observed power in the corresponding cluster for better resolution and visualization. The difference in the overall rCRPS score of all the models is statistically insignificant, with all models close to 0.475. However, it is evident that GEKKO performs significantly better compared with the two other models for $K_t \geq 0.7$, whereas weighted average ensemble has better CRPS for $K_t < 0.7$, and the statistical difference between weighted average ensemble and GB is insignificant.

The rCRPS of the three models is shown as a function of the time of the day in Figure 12b. The rCRPS of GEKKO is significantly lower for 70% of the hours of the day.

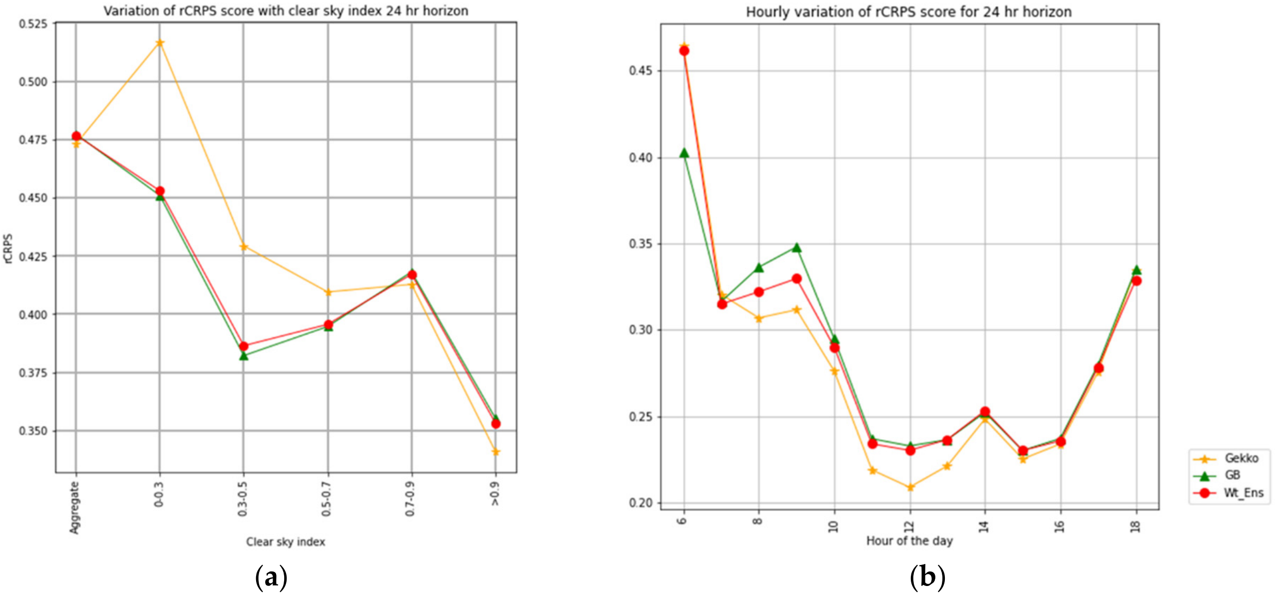


Figure 12. Relative CRPS score: (**a**) aggregate and variation with K_t; (**b**) hourly variation.

Although the models under evaluation forecast a continuous variable, BS is used to compare the performance of the models to predict common and rare events. In this study, BS is calculated for two rare events and one normal event:

- Rare event R1, where K_t is lower than its 10% percentile;
- Rare event R2, where K_t is higher than its 90% percentile;
- Normal event N1, where K_t is higher than its 50% percentile.

The threshold value is kept at 0.5 of the maximum power value in each cluster for BS evaluation, as per the standard practice.

Overall values of BS for three events, as shown in Figure 13, show statistically significant differences for the three models. GEKKO is more skilled for the normal event, N1, as well as for higher K_t event R2. However, the performance of GEKKO drops significantly for R1, where GB seems to perform better, suggesting it is better at predicting extreme low power to some degree.

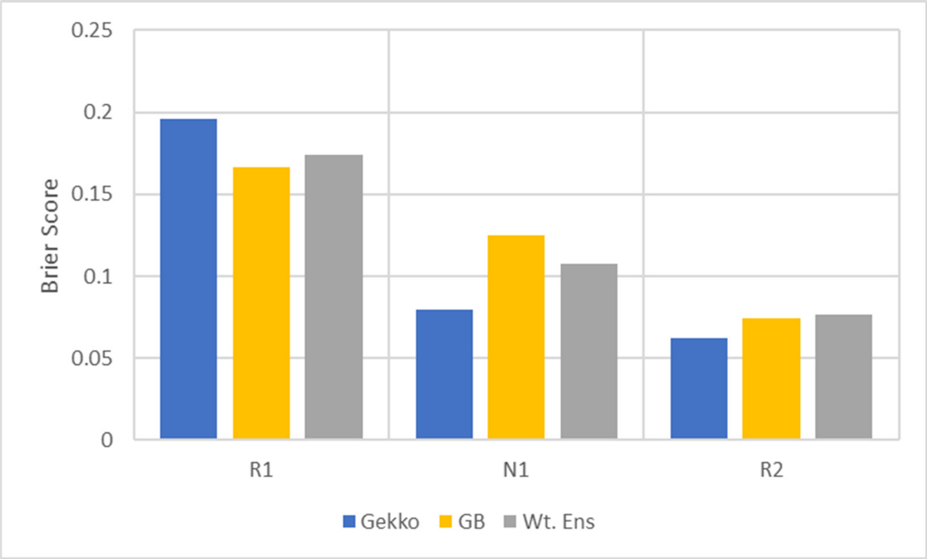


Figure 13. Brier scores for three events.

The BS of the three models is shown as a function of the time of day for each event in Figure 14. GEKKO is apparently more skilled for common event N1, and the difference between GEKKO and weighted average ensemble is significant for 54% of the hours of the day, from 11:00 to 17:00 h. The performance of GEKKO is mostly lower compared to the other models for event R1; however, its performance improves significantly for event R2 for higher power hours of the day, from 11:00 to 15:00 h.

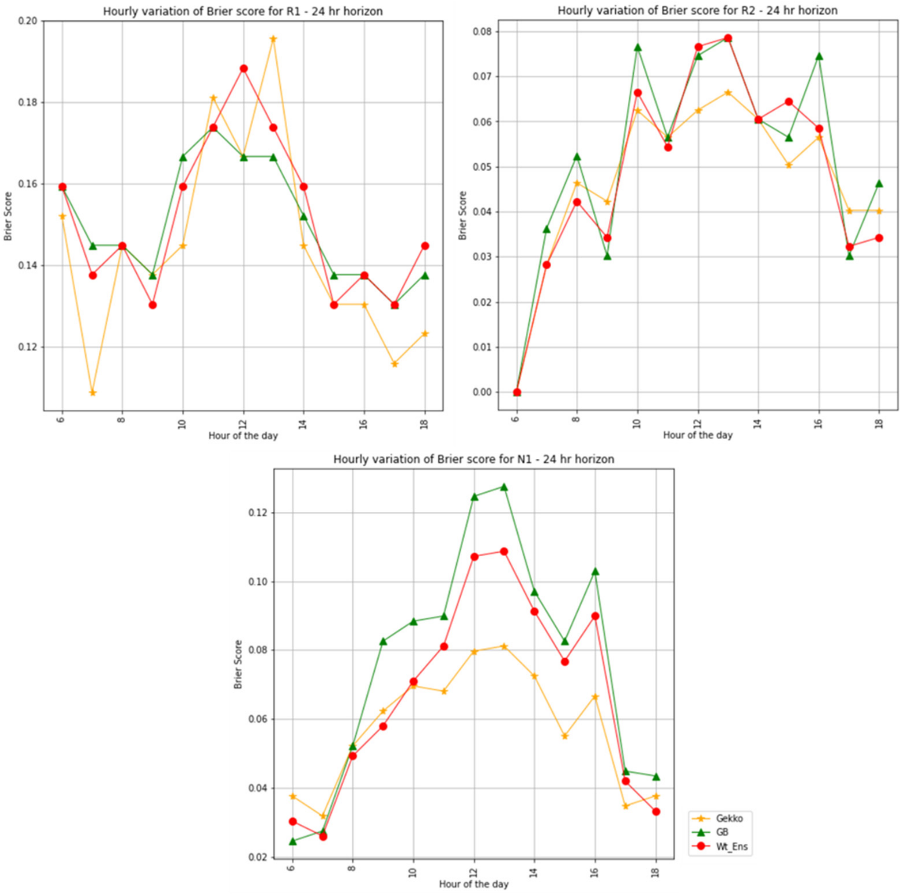


Figure 14. Hourly variation of Brier scores for three events.

5. Conclusions and Outlook

The accurate and reliable forecasting of PV energy production has gained tremendous importance and dynamics in recent years from both a research and economic point of view due to a rapidly growing integration of solar energy into the power grid. In this study, the prediction of PV energy based on meteorological variables is presented, which are available from meteorological data providers. A study was performed to evaluate the impact of different cloud conditions by classifying the clear sky index into clusters to enhance the forecast performance. As benchmark models, several linear regression models and time series models were applied and compared to evaluate the possible gains in applying computationally more demanding models. Several ML models were used and compared for both deterministic and probabilistic forecasts because solar power possesses nonlinear properties.

In this paper, we also proposed a novel method based on GEKKO-optimized polynomial regression forecast methodology. The overall results show that the proposed GEKKO-based methodology works better than any other model, even under central European cloudy conditions. The forecasting accuracy can be further increased for tropical countries and southern Europe because of the large proportion of clear sky data in such datasets.

Further improvement in GEKKO forecast is possible by optimizing regressive coefficients under cloudy and overcast sky conditions with larger datasets. This methodology is also verified and applied for other PV technologies, such as copper indium selenide (CIS), amorphous silicon (a-Si), cadmium telluride (CdTe) and polycrystalline silicon (p-Si) and can be adapted for any location. The forecasting performance beyond a 24-h horizon can be improved by optimizing the GEKKO model to separately derive coefficients for respective forecast horizons. Different ML models show significant improvements, especially under cloudy to overcast sky conditions. The quantile loss study presented in Section 4.2 can be further extended in quantile gradient boosting (QGB) and other ML models, such as analog ensemble models, to build a more robust probabilistic procedure. Forecasting accuracy can be further leveraged in QGB by penalizing the model for overestimating or underestimating, thereby enhancing the forecasting accuracy. This study shows the potential of applying various ML methods with limited input and logical clustering methodology but nonetheless with promising forecasting results for PV energy.

Author Contributions: Conceptualization, J.T. and R.H.; formal analysis, J.T.; software, J.T.; visualization, J.T.; validation, J.T.; writing—original draft preparation, J.T.; writing—review and editing, R.H.; project administration, R.H.; funding acquisition, R.H.; supervision, R.H. All authors have read and agreed to the published version of the manuscript.

Funding: This project has received funding from the European Union's Horizon 2020 Research and Innovation program under Grant Agreement No. 957755.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Acknowledgments: Open Access Funding by the University for Continuing Education Krems, the University of Applied Sciences BFI Vienna and the University of Applied Sciences Upper Austria. We thank Blue Sky Wetteranalysen for their consent to use the forecasting data of GHI and temperature for this study. We also wish to thank Luca Massidda for valuable advice and Germanno Longhi Beck for implementing the models on servers to render operational forecasts for the SENDER project.

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

Abbreviations

PV	Photovoltaics
GHI	Global Horizontal Irradiance
K_t	Clear Sky Index
ML	Machine Learning
ANN	Artificial Neural Network
GFS	Global Forecast System
ECMWF	European Centre for Medium Range Forecasts
NAM	North American Mesoscale Model
WRF	Weather Research Forecast
NWP	Numerical Weather Prediction
ARIMA	Auto Regressive Integrated Moving Average
SARIMAX	Seasonal Auto Regressive Integrated Moving Average Exogenous
RR	Ridge Regression
LASSO	Least Absolute Shrinkage Selector Operator
EN	Elastic Net
FBP	Facebook Prophet
RF	Random Forest
GB	Gradient Boosting
XGB	Extreme Gradient Boosting
FHOOE	Fachhochschule Oberösterreich
CIS	Copper Indium Selenium
CdTe	Cadmium Telluride
a-Si	Amorphous Silicon
p-Si	Polycrystalline Silicon
c-Si	Monocrystalline Silicon
MBE	Mean Bias Error
MAE	Mean Absolute Error
rMAE	Relative Mean Absolute Error
nMAE	Normalized Mean Absolute Error
IMAE	Inverse of Mean Absolute Error
RMSE	Root Mean Square Error
rRMSE	Relative Root Mean Square Error
nRMSE	Normalized Root Mean Square Error
CRPS	Continuous Ranking Probability Score
rCRPS	Relative Continuous Ranking Probability Score
PIT	Probability Integral Transform
BS	Brier Score

References

1. Massidda, L.; Marrocu, M. Decoupling Weather Influence from User Habits for an Optimal Electric Load Forecast System. *Energies* **2017**, *10*, 2171. [CrossRef]
2. D'Alessandro, V.; Di Napoli, F.; Guerriero, P.; Daliento, S. An automated high granularity tool for a fast evaluation of the yield of PV plants accounting for shading effects. *Renew. Energy* **2015**, *83*, 294–304. [CrossRef]
3. Monteiro, C.; Fernandez-Jimenez, L.A.; Ramirez-Rosado, I.J.; Muñoz-Jimenez, A.; Lara-Santillan, P.M. Short-term forecasting models for photovoltaic plants: Analytical versus soft-computing techniques. *Math. Probl. Eng.* **2013**, *2013*, 767284. [CrossRef]
4. Das, U.K.; Tey, K.S.; Seyedmahmoudian, M.; Mekhilef, S.; Idris, M.Y.I.; Deventer, W.V.; Horan, B.; Stojcevski, A. Forecasting of photovoltaic power generation and model optimization: A review. *Renew. Sustain. Energy Rev.* **2018**, *81*, 912–928. [CrossRef]
5. Monteiro, C.; Santos, T.; Fernandez-Jimenez, L.A.; Ramirez-Rosado, I.J.; Terreros-Olarte, M.S. Short-term power forecasting model for photovoltaic plants based on historical similarity. *Energies* **2013**, *6*, 2624–2643. [CrossRef]
6. Yang, D.; Wang, W.; Gueymard, C.A.; Hong, T.; Kleissl, J.; Huang, J.; Perez, M.J.; Perez, R.; Bright, J.M.; Xia, X.; et al. A review of solar forecasting, its dependence on atmospheric sciences and implications for grid integration: Towards carbon neutrality. *Renew. Sustain. Energy Rev.* **2022**, *161*, 112348. [CrossRef]
7. Box, G.E.; Jenkins, G.M. Some recent advances in forecasting and control. *J. R. Stat. Soc. Ser. C Appl. Stat.* **1968**, *17*, 91–109. [CrossRef]
8. Salinas, D.; Flunkert, V.; Gasthaus, J. DeepAR: Probabilistic Forecasting with Autoregressive Recurrent Networks. *Int. J. Forecast.* **2020**, *36*, 1181–1191. [CrossRef]

9. Antonanzas, J.; Osorio, N.; Escobar, R.; Urraca, R.; Martinez-De-Pison, F.; Antonanzas-Torres, F. Review of photovoltaic power forecasting. *Sol. Energy* **2016**, *136*, 78–111. [CrossRef]
10. Montgomery, D.C.; Jennings, C.L.; Kulahci, M. *Introduction to Time Series Analysis and Forecasting*, 1st ed.; Wiley: Hoboken, NJ, USA, 2008.
11. Reikard, G. Predicting solar radiation at high resolutions: A comparison of time series forecasts. *Sol. Energy* **2009**, *83*, 342–349. [CrossRef]
12. Raza, M.Q.; Nadarajah, M.; Ekanayake, C. On recent advances in PV output power forecast. *Sol. Energy* **2016**, *136*, 125–144. [CrossRef]
13. Theocharides, S.; Makrides, G.; Livera, A.; Theristis, M.; Kaimakis, P.; Georghiou, G.E. Day-ahead photovoltaic power production forecasting methodology based on machine learning and statistical post-processing. *Appl. Energy* **2020**, *268*, 115023. [CrossRef]
14. Visser, L.; AlSkaif, T.; van Sark, W. Operational day-ahead solar power forecasting for aggregated PV systems with a varying spatial distribution. *Renew. Energy* **2022**, *183*, 267–282. [CrossRef]
15. Natschläger, T.; Traunmüller, W.; Reingruber, K.; Exner, H. Lokal optimierte Wetterprognosen zur Regelung stark umweltbeeinflusster Systeme; SCCH, Blue Sky. In *Tagungsband Industrielles Symposium Mechatronik Automatisierung*; Clusterland Oberösterreich GmbH/Mechatronik-Cluster: Wels, Austria, 2008; pp. 281–284.
16. Bergmeir, C.; Benietez, J.M. On the use of cross-validation for time series predictor evaluation. *Inf. Sci.* **2011**, *191*, 192–213. [CrossRef]
17. Zhang, J.; Hodge, B.-M.; Florita, A.; Lu, S.; Hamann, H.F.; Banunarayanan, V. Metrics for Evaluating the Accuracy of Solar Power Forecasting. In Proceedings of the 3rd International Workshop on Integration of Solar Power into Power Systems, London, UK, 21–22 October 2013.
18. Lorenz, E.; Heinemann, D. Prediction of solar irradiance and photovoltaic power. In *Comprehensive Renewable Energy*; Sayigh, A., Ed.; Elsevier: Oxford, UK, 2012; pp. 239–292.
19. Reindl, T.; Walsh, W.; Yanqin, Z.; Bieri, M. Energy meteorology for accurate forecasting of PV power output on different time horizons. *Energy Procedia* **2017**, *130*, 130–138. [CrossRef]
20. Coimbra, C.F.M.; Kleissl, J.; Marquez, R. Overview of solar-forecasting methods and a metric for accuracy evaluation. In *Solar Energy Forecasting and Resource Assessment*; Kleissl, J., Ed.; Academic Press: Oxford, UK, 2013; pp. 171–194.
21. Wilks Daniel, S. *Statistical Methods in The Atmospheric Sciences*, 3rd ed.; Academic Press: Cambridge, MA, USA, 2011; ISBN 9780123850232.
22. Pinson, P.; Nielsen, H.A.; Muller, J.K.; Madsen, H.; Kariniotakis, G.N. Non-parametric probabilistic forecasts of wind power: Required properties and evaluation. *J. Prog. Appl. Wind. Power Convers. Technol.* **2007**, *10*, 497–516. [CrossRef]
23. Available online: https://en.wikipedia.org/wiki/Probability_integral_transform#cite_note-Dodge-1 (accessed on 17 December 2021).
24. Gneiting, T.; Raftery, A.E. Strictly proper scoring rules, prediction, and estimation. *J. Am. Stat. Assoc.* **2007**, *102*, 359–378. [CrossRef]
25. Massidda, L.; Marrocu, M. Quantile Regression Post-Processing of Weather Forecast for Short-Term Solar Power Probabilistic Forecasting. *Energies* **2018**, *11*, 1763. [CrossRef]
26. Holmgren, W.F.; Hansen, C.W.; Mikofski, M.A. PVLIB python: A python package for modeling solar energy systems. *J. Open Source Softw.* **2018**, *3*, 884. [CrossRef]
27. Brockwell, P.J.; Davis, R.A. *Time Series: Theory and Methods*, 2nd ed.; Springer Series in Statistics; Springer: Berlin/Heidelberg, Germany, 2009.
28. Kuo, P.-H.; Huang, C.-J. A Green Energy Application in Energy Management Systems by an Artificial Intelligence-Based Solar Radiation Forecasting Model. *Energies* **2018**, *11*, 819. [CrossRef]
29. Available online: https://scikit-learn.org/stable/modules/linear_model.html (accessed on 12 February 2021).
30. Hilt, D.E.; Seegrist, D.W. *Ridge, A Computer Program for Calculating Ridge Regression Estimates*; Department of Agriculture, Forest Service, Northeastern Forest Experiment Station: Amherst, MA, USA, 1977; Available online: https://www.nrs.fs.fed.us/pubs/9260 (accessed on 20 February 2021).
31. Dickey, D.A.; Fuller, W.A. Distribution of the Estimators for Autoregressive Time Series with a Unit Root. *J. Am. Stat. Assoc.* **1979**, *74*, 427–431.
32. Suhartono, S.; Rahayu, S.P.; Prastyo, D.; Wijayanti, D.G.P. Hybrid model for forecasting time series with trend, seasonal and calendar variation pattern. *J. Phys. Conf. Ser.* **2017**, *890*, 012160. [CrossRef]
33. Taylor, S.J.; Letham, B. Forecasting at scale. *Am. Stat.* **2018**, *72*, 37–45. [CrossRef]
34. Leo, B.; Adele, C. Random Forests. Available online: https://www.stat.berkeley.edu/~{}breiman/randomforest2001.pdf (accessed on 15 October 2021).
35. Kam, H.T. Random decision forests. In Proceedings of the 3rd International Conference on Document Analysis and Recognition, Montreal, QC, Canada, 14–16 August 1995; Volume 1416, pp. 278–282.
36. Leo, B. Bagging Predictors. *Mach. Learn.* **1996**, *24*, 123–140.
37. Available online: https://www.ibm.com/cloud/learn/random-forest (accessed on 8 June 2021).
38. Available online: https://en.wikipedia.org/wiki/Random_forest (accessed on 8 June 2021).
39. Hastie, T.; Tibshirani, R.; Friedman, J. *The Elements of Statistical Learning*, 2nd ed.; Springer: Berlin/Heidelberg, Germany, 2008.

40. Freund, Y.; Schapire, R.E. Experiments with a new boosting algorithm. In Proceedings of the 13th International Conference on Machine Learning, Bari, Italy, 3–6 July 1996.
41. Friedman, J.; Hastie, T.; Tibshirani, R. Additive logistic regression: A statistical view of boosting (with discussion and a rejoinder by the authors). *Ann. Stat.* **2000**, *28*, 337–407. [CrossRef]
42. Friedman, J.H. Greedy function approximation: A gradient boosting machine. *Ann. Stat.* **2001**, *29*, 1189–1232. [CrossRef]
43. Bogner, K.; Pappenberger, F.; Zappa, M. Machine Learning Techniques for Predicting the Energy Consumption/Production and its Uncertainties Driven by Meteorological Observations and Forecasts. *Sustainability* **2019**, *11*, 3328. [CrossRef]
44. Verbois, H.; Rusydi, A.; Thiery, A. Probabilistic forecasting of day-ahead solar irradiance using quantile gradient boosting. *Sol. Energy* **2018**, *173*, 313–327. [CrossRef]
45. Zou, H.; Hastie, T. Regularization and variable selection via the elastic net. *J. R. Stat. Soc. Ser. B Stat. Methodol.* **2005**, *67*, 301–320. [CrossRef]
46. 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 (KDD '16), New York, NY, USA, 13 August 2016; pp. 785–794.
47. Available online: https://bradleyboehmke.github.io/HOML/gbm.html (accessed on 8 September 2021).
48. Beal, L.D.R.; Hill, D.; Martin, R.A.; Hedengren, J.D. GEKKO Optimization Suite. *Processes* **2018**, *6*, 106. [CrossRef]
49. Available online: https://gekko.readthedocs.io/en/latest/overview.html (accessed on 8 February 2021).
50. Hedengren, J.D.; Asgharzadeh, S.R.; Powell, K.M.; Edgar, T.F. Nonlinear Modeling, Estimation and Predictive Control in APMonitor. *Comput. Chem. Eng.* **2014**, *70*, 133–148. [CrossRef]
51. Ferruzzi, G.; Cervone, G.; Monache, L.D.; Graditi, G.; Jacobone, F. Optimal bidding in a Day-Ahead energy market for Micro Grid under uncertainty in renewable energy production. *Energy* **2016**, *106*, 194–202. [CrossRef]
52. Alessandrini, S.; Davo, F.; Sperati, S.; Benini, M.; Luca, D.M. Comparison of the economic impact of different wind power forecast systems for producers. *Adv. Sci. Res.* **2014**, *11*, 49–53. [CrossRef]
53. Nau, R. Review of Basic Statistics and the Simplest Forecasting Model. Available online: https://people.duke.edu/~{}rnau/review_of_basic_statistics_and_the_mean_model_for_forecasting--robert_nau.pdf (accessed on 15 June 2021).
54. Kriegler, B.; Berk, R. Boosting the Quantile Distribution: A Cost-Sensitive Statistical Learning Procedure. 2007. Available online: https://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.103.6688&rep=rep1&type=pdf (accessed on 10 September 2021).