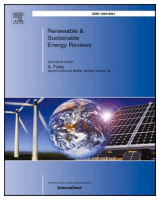
Renewable and Sustainable Energy Reviews 161 (2022) 112364

Contents lists available at ScienceDirect

Renewable and Sustainable Energy Reviews

journal homepage: www.elsevier.com/locate/rser



Comparison of machine learning methods for photovoltaic power forecasting based on numerical weather prediction

Dávid Markovics, Martin János Mayer *

Department of Energy Engineering, Faculty of Mechanical Engineering, Budapest University of Technology and Economics, Műegyetem rkp. 3, H-1111 Budapest, Hungary

ARTICLE INFO

Keywords:
Solar power forecasting
Machine learning
Photovoltaic power production
Irradiance-to-power conversion
Hyperparameter tuning
Kernel ridge regression
Multilayer perceptron
Predictor selection

ABSTRACT

The increase of the worldwide installed photovoltaic (PV) capacity and the intermittent nature of the solar resource highlights the importance of power forecasting for the grid integration of the technology. This study compares 24 machine learning models for deterministic day-ahead power forecasting based on numerical weather predictions (NWP), tested for two-year-long 15-min resolution datasets of 16 PV plants in Hungary. The effects of the predictor selection and the benefits of the hyperparameter tuning are also evaluated. The results show that the two most accurate models are kernel ridge regression and multilayer perceptron with an up to 44.6% forecast skill score over persistence. Supplementing the basic NWP data with Sun position angles and statistically processed irradiance values as the inputs of the learning models results in a 13.1% decrease of the root mean square error (RMSE), which underlines the importance of the predictor selection. The hyperparameter tuning is essential to exploit the full potential of the models, especially for the less robust models, which are prone to under or overfitting without proper tuning. The overall best forecasts have a 13.9% lower RMSE compared to the baseline scenario of using linear regression. Moreover, the power forecasts based on only daily average irradiance forecasts and the Sun position angles have only a 1.5% higher RMSE than the best scenario, which demonstrates the effectiveness of machine learning even for limited data availability. The results of this paper can support both researchers and practitioners in constructing the best data-driven techniques for NWP-based PV power forecasting.

1. Introduction

Sustainability, reliability, and flexibility – some buzzwords that can be found in all energy strategy in the dawn of the 2020-s. According to the EU energy strategy [1], photovoltaic (PV) power production plays an important role in accomplishing the various climate protection goals. On the other hand, the fast-growing installed capacity [2] of this weather-dependent technology has brought along some serious challenges for Transmission and Distribution System Operator (TSO and DSO) professionals at balancing and grid integration, which highlighted the importance of the possibly most accurate forecasting techniques. Consequently, solar forecasting research has been flourishing in the last decade, and it has almost caught up with such prominent topics of energy forecasting as wind and load forecasting in terms of the number of publications [3]. A large number of studies were published in the last

Abbreviations: AE, Average Error; AR, Autoregressive; BgR, Bagging Regression; CSTF, Compressive Spatio-Temporal Forecasting; CSP, Clear Sky Persistence; DTR, Decision Tree Regression; ELM, Elman Backpropagation Network; ETS, ETS stands for Error-Trend-Seasonality, often referred as exponential smoothing; FNN, Feed Forward Network; GHI, Global Horizontal Irradiance; GLSSVM, Group Least Square Support Vector Machine, a novel hybrid algorithm based on the combination of LS-SVM and GMDH models; GMDH, Group Method of Data Handling) a relatively unexplored neural network; GRU, Gated Recurrent Unit; KSI, Kolmogorov–Smirnov test integral; KURT, Kurtosis describes the magnitude of the peak or the width of the distribution of forecast errors; LS-based ST, Least Square based Spatio-Temporal model; LS-SVM, Least Square Support Vector Machine; LSTM, Long Short-Term Memory; MAPE, Mean Absolute Percentage Error; ME, Maximum Error; ML, Machine Learning; MPIW, Mean Prediction Interval Width; MR, Multiple Regression; nE, Normalized error; NewCF, Cascade-Forward Backpropagation Network; nMBE, Normalized Mean Bias Error; nMSE, Normalized Mean Square Error; nRMSE, Normalized Root Mean Square Error; OPSD, Open Power System Data; PAR, photosynthetic active radiation; PM, Persistence Model; RNN, Recurrent Neural Network; RE, Relative Error; SARIMA(X), Seasonal Auto-Regressive integrated Moing Average (with eXogenous factors); SDE, Standard Deviation Error; SKEW, Skewness, a measure of the asymmetry of the error probability distribution; SPM, Smart Persistence model; SVR, Support Vector Regression; WMAE, Weighted Mean Absolute Error; WPD, Wavelet Packet Decomposition; WT-ANN, Wavelet Transform Artificial Neural Network.

* Corresponding author.
E-mail address: mayer@energia.bme.hu (M.J. Mayer).

https://doi.org/10.1016/j.rser.2022.112364
Received 6 October 2021; Received in revised form 16 December 2021; Accepted 5 March 2022
Available online 23 March 2022

List of notations

ρ	Pearson correlation coefficient
σ	standard deviation
$cov()$	covariance
f_i	predicted output power
i	number of the given timestep
N	length of the daytime dataset's 15-min intervals
s	skill score
Ta	ambient (air) temperature
Ws	wind speed

decade to introduce novel and promising methods for solar forecasting for different locations and time horizons [4]. Still, the conclusions of the Global Energy Forecasting Competition 2014 revealed that solar forecasting was highly immature compared to other domains of energy forecasting [5]. As a reason, Yang identified that many solar forecasting papers have only a limited added value due to issues like possible duplications and limited reproducibility and comparability [6]. Even though many different methods are available for almost any kind of solar forecasting task, it is rarely known which one is the most effective in a certain practical application.

Consequently, the time has come for the solar forecasting community to focus on consolidating the existing knowledge instead of just chasing further seemingly more accurate forecasting methods. A notable example is a recent paper authored by 33 prominent researchers of the field that aim to formulate a standardized way for the verification of deterministic solar forecasts [7]. Another important milestone is a recent review that classified post-processing techniques into ten thinking tools instead of using the traditional but less effective method-based classification [8]. Furthermore, a tendency is formulating in recent years to publish papers that compare a large number of different methods for the same datasets for a given task, e.g., machine learning models for hourly solar irradiance forecasting [9] or physical models for irradiance-to-power forecast conversion [10]. The importance and value of such studies originate from the limited cross-scenario comparability of solar forecast accuracy. Blaga et al. [11] used mean-normalized mean bias error (nMBE) and root mean square error (nRMSE) for the comparison of a large number of forecasting studies to discover how forecast accuracy depends on the forecast horizon and six main model classes. However, as the predictability of the solar irradiance depends on its variability, not even the normalized forms of the common error metrics are suitable for comparing forecasting methods verified for different locations and timescales [7]. As a partial solution, skill scores have long been used in weather forecasting to quantify the improvement over a naïve reference method [12]. In solar forecasting, the most commonly used reference method is clear-sky persistence [13], while recent studies argue for using the optimal convex combination of climatology and persistence as the standard reference [14]. Using the most accurate naïve method is good to avoid exaggerating the forecast skill, but it hinders the comparability of skill scores calculated over different references. Even though skill scores are considered the best metrics to measure the skillfulness of forecasters, they are not able to reliably compare different methods, as there is no guarantee that the same method will have exactly the same skill score for different datasets [10]. Consequently, it is almost impossible to find the more accurate of two distinct methods if they were evaluated for different locations, time horizons, and temporal resolutions unless reproducing and verifying them for the same dataset.

The largest part of solar forecasting research deals with irradiance forecasting, which requires a further step of irradiance-to-power conversion to create PV power forecasts with an economic value [15]. The two distinct approaches used for irradiance-to-power conversion are physical and statistical, which are also referred to as indirect and direct, respectively [8]. A third category, namely the hybrid, is often distinguished, which is a general category referring to the hybridization of any two physical or statistical methods [16]. The physical approach relies on a so-called model chain that consists of several steps, like beam and diffuse separation, tilted irradiance transposition, and PV efficiency modeling. Even though physical model chains have long been used for PV power forecasting [17], a recent paper was the first that discussed the effect of model selection on the power forecast accuracy and found that it has a considerable impact [10]. Physical modeling requires detailed knowledge of the PV plant to achieve its possible best accuracy; however, as a recent study has shown, it also has a decent accuracy even if only the orientation of the PV modules and the nominal power of the plant is known [18].

The statistical approach includes all data-driven methods ranging from classical statistical to the newest machine learning models. These models are based on historical data for training, but they do not require any knowledge related to the design parameters of the PV plants, which makes them easy to implement in many practical applications. Accordingly, Antonanzas et al. [19] found that machine learning models are most commonly used for PV power forecasting. An overview of studies using data-driven irradiance-to-power conversion methods is presented in the next subsection.

1.1. Summary of statistical irradiance-to-power conversion studies

Table 1 summarizes several notable papers that use statistical methods for photovoltaic power forecasting. The most common source of day-ahead irradiance forecasts is numerical weather prediction (NWP). In some cases, the main test methods are followed by brackets in the first and second columns listing those techniques that have been used only for benchmarking.

The most used methods are arguably the neural network algorithms, followed by support vector regression. Regarding evaluation, there is no general accordance in the methodology, as a wide range of error metrics have been used in the examined papers. Only RMSE and its normalized versions stand out of the list, as they have been the most frequently applied. However, the base of normalizations can differ between average [20], rated, or measured maximum power [21–24]. In terms of the time horizon, mostly day-ahead and some shorter experiments have been done. As observed by Yang et al. [15], the lead time of the applied forecast is rarely clarified.

1.2. Contribution of this paper

As machine learning has long been used for modeling PV plants [38], statistical irradiance-to-power conversion methods have high relevance in practical applications like scheduling PV plant power output for the day-ahead market. Solar irradiance forecasts can be accessed from meteorological agencies or other commercial service providers; therefore, PV plant owners or operators should only convert the irradiance to power forecasts. With the lack of knowledge about the exact specifications of distributed PV parks, in the case of TSO-s, data-driven methods are prominently relevant. With such user-friendly packages as *scikit-learn* or *PyCaret* for Python and *caret* for R, machine learning algorithms can be easily used by both researchers and practical experts even without deep theoretical data science knowledge.

However, no reliable recommendations can yet be found in the literature regarding the model, predictor, or hyperparameters selection. Similarly to the physical approach, where the model selection has a significant effect on the power forecast accuracy [10], the choice and fine-tuning of the machine learning model is also expected to be important. This study aims to fill this research gap by comparing 24 easily accessible machine learning methods for NWP-based deterministic day-ahead PV power forecasting for 2-year long datasets from 16 PV plants, following the latest recommendations for the solar forecast

Table 1
Summary of recent photovoltaic power forecasting studies.

Study	Nr. of tested methods	Tested methods	Metrics	Horizon	Data resolution	Data length	Predictors	Highlights of study
[25]	12	MLR, SARIMAX, LASSO, SVM, RF, GBR, ANN, LSTM, Physical model, PM, CSP	MAE, RMSE, MBE, Economic revenues	DA	1 h	3 years	pressure (air, mean sea level), temperature (ambient, dewpoint), precipitation, wind speed (zonal, meridional), cloud cover variations, clear sky GHI, surface solar radiation, zenith and azimuth angle, sine and cosine of hour of day	Compared 12 forecasting methods for hourly day-ahead power forecasting of single and aggregated systems based on NWP, considering the market (DAM) requirements. The results were evaluated from both technical and financial perspective, highlighting the controversiality of choosing the optimal methods by different objectives.
[26]	3 (4)	GRU (3), (WPD-SVM, RF, LSTM, CNN)	nMAE, nRMSE	DA	5-min	3 years	historical solar power	Proposed a deep learning based temporal distributed hybrid model for day-ahead PV generation forecasting of distributed power plants without which are not supported by local weather station, reaching the accuracy level of NWP-based models.
[27]	6	MR (3), ANN (3)	MAE, MSE, RMSE, Error, R^2	DA	15-min	3 years	measured temperature, wind speed, cloud opacity, dewpoint temperature, irradiance components, precipitation, humidity, snow depth, PV power and calculated cell temperature	Analyzed ANN and MR techniques for day-ahead power forecasting on 3 years of historical measured data of a 546 kWp PV park, proving that ANN models perform better regardless of input method.
[28]	10	DTR, LR, K-NN, RF, ABR, GBR, BgR, PSO-LSTM, PSO-BI-LSTM, PSO-SVR	MAE, RMSE, R^2	DA	30-min	2 years (1-1)	GARV, precipitation, ambient temperature, panel temperature, wind direction, wind speed, sunshine duration	The first study that used automatic machine learning to find the best ensemble model, also increased accuracy by using operators to reconstruct meteorological factors selected by an improved GA algorithm, tested on multi-region PV parks.
[29]	1 (3)	LSTM-RNN, (BPNN, SVM, PM)	RMSE, MAE, Corr (normalized)	DA	15-min	5 years	measured PV power in the previous three days	Established an independent LSTM-RNN model for day-ahead PV power forecasting and tested a modification method to update the predictions based on time correlation principles regarding daily power patterns, for which a partial daily pattern prediction (PDPP) framework was proposed successfully.
[30]	1	LSTM architecture	MAPE, RMSE	DA	10-min	1 month	temperature, time of day	Applied different LSTM-based architectures to generate PV power forecast with a low amount of previous data (31 days).
[31]	4	SVM, MLP, MARS, ensemble	RMSE, MAE, MAPE	–	5-min	~5 years	wind speed, temperature, relative humidity, global horizontal irradiation, diffuse horizontal radiation, and wind direction	Built an ensemble of stand-alone models with a recursive process to optimize weights using measured weather data from Desert Knowledge Australia Solar Centre (DKASC).

(*continued on next page*)

Table 1 (*continued*)

Study	Nr. of tested methods	Tested methods	Metrics	Horizon	Data resolution	Data length	Predictors	Highlights of study
[9]	6 families (in total: 142 combinations)	SARIMA family of models (36), ETS family of models (30), MLP (1), STL decomposition (2), TBATS, Theta model	nRMSE, nMBE, SKILL, KSI	DA	1 h	350 days	–	Tested several time series ensembles from six model families on power and NWP forecast data from multiple PV parks, proving that forecast combinations are less risky than stand-alone models.
[32]	1 (8)	HKGE (RBFNN, BPNN, LS-SVM, ELM, HGB, HGR, HGL, HGE)	RMSE, MAPE, R^2	–	15-min	2 years	global and diffuse horizontal irradiation, air temperature, relative humidity, wind speed	Introduced a hybrid forecasting model combining improved K-means clustering, grey relational analysis and Elman neural network for short-term prediction, using measured weather data from DKASC.
[33]	4 (1)	FNN, ELM, NewCF, BMA (PM)	R^2, MAPE	DA	1-min	2 years	solar irradiance, PV module temperature, humidity, wind speed, average of solar irradiance	Created a multivariate ensemble framework of FNN, ELM and NewCF models with additional BMA method for combining predictors, tested for day-ahead forecasting using data from different PV parks.
[24]	5	ANN, LS-SVM, GPR, WGPR (LOF), WGPR (WLOF)	AE, ME, RMSE, WMAE, nMSE, nRMSE, MPIW, Time	5-min ahead	5-min	180 days	PAR, ambient temperature, relative humidity, wind speed, wind direction, solar radiation, and precipitation	Integrated outlier detection method (WGPR) to improve regression model to reduce the effect of high outlier potential datapoints, tested for short-term forecasting using measured weather data.
[34]	1	ANN	PM, SPM, Forecast error, RMSE, MAE, MBE, Skill score, Corr	DA	20-min	2 years	azimuth, elevation, clear sky GHI, ground temperature, RH of 20 vertical atmospheric levels	Analyzed the effect of growing PV penetration to transmission system imbalance by applying ANN methods for day-ahead power forecasting from NWP and actual production data. The results were financially evaluated and compared to then actually used reference models.
[21]	5	Markov chain and space fusion with weather type modifications	RMSE, nMAE	DA	15-min	2 years	–	Proposed a space fusion model based on historical similar-day categorization, weather-type classification from forecast information, modified by seasonal Markov chain state transition probability matrixes, and persistence model, using incomplete data.
[35]	1	SVM (GA optimization)	RE	DA	15-min	–	temperature, weather humidity, weather type	Introduced a day-ahead forecasting method using SVM, with GA optimized parameters and weather type classification of NWP data by grey correlation coefficient algorithm.
[23]	1	GLSSVM (5 preprocessing variation)	nE, nMBE, nRMSE, SKEW, Kurt	1-h to 24 h	1-h	1 year	ambient temperature, solar irradiance on tilted of array, wind speed	Presented the forecast performance of GLSSVM with different data-preprocessing techniques on several time horizons from 1 to 24 h, using measured weather data.

(*continued on next page*)

Table 1 (*continued*)

Study	Nr. of tested methods	Tested methods	Metrics	Horizon	Data resolution	Data length	Predictors	Highlights of study
[22]	3	LS-SVM, GLSSVM, GMDH (with 3 type of strategies)	nE, nMBE, nRMSE, SKEW, Kurt, SDE	1-h to 24 h	1-h	1.75 years	module temperature, ambient temperature, irradiance, previous power values	Compared Direct, Recursive and DirRec strategies for multi-step ahead power forecasting using hybrid GLSSVM and stand-alone reference methods like GMDH and LS-SVM.
[36]	1 (5)	Nonuniform CSTF (PM, AR, WT-ANN, Multi-input ANN, LS-based ST)	MAE, RMSE, nRMSE, Skill	3-h	15-min	1 year	irradiance on the tilted plane, ambient temperature, wind speed	Proposed a spatiotemporal forecasting model called Nonuniform CSTF, which outperformed the neural network methods used as benchmarks on 3-h-ahead prediction, highlighting the importance of spatial correlation, especially in short-term horizon.
[37]	2	SVR, physical	MBE, RMSE	15-min to 5-h	15-min	2 years	PV power measurements, numerical weather prediction (NWP) and cloud motion vector (CMV) irradiance forecasts	Investigated the effect of input data and model selection by optimizing an SVR algorithm with different predictor combinations for short-term forecasting and comparing the performance to a physical model and a linear regression-based hybrid combination of the methods.
This paper	24	LR, Lasso, Ridge, EN, Lars, OMP, BR, ARD, PAR, RANSAC, TR, Huber, KR, SVM, MLP, KNN, DT, RF, ET, ABR, GBR, XGBoost, LGBM, CatBoost	nMAE, nRMSE, Skill score, Corr, Time	DA	15-min	2 years	GHI, ambient temperature, wind speed, elevation, azimuth and declination angle	

verification. The results of the presented analysis can serve as a guide for the model and hyperparameter selection for PV power forecasting in a wide range of practical and research applications.

In Section 2, the methodology of this paper is introduced, starting with the power production data and the explanatory variables, followed by the used machine learning techniques, hyperparameter tuning process, and verification metrics. Section 3 contains the results and discussion, starting with a general presentation of the results for all the different cases, which are further investigated from different aspects, like the error metrics, hyperparameter tuning, and visual comparison. The conclusions and recommendations are summarized in Section 4.

2. Materials and methods

This section presents the meteorological and PV power production data, the machine learning models, the hyperparameters turning process, and the metrics used for the verification.

2.1. Data

Reliable, high-quality, and long-term historical data are crucial for machine learning applications. However, the availability of such data is often limited due to the financial conflict of interest in the triangle of power producers with the regulative need to schedule, the meteorological forecast suppliers, and the sovereign service companies. The data used in this research are completely substantive, provided by Hungarian industrial members. Due to the annual periodicity of weather patterns, at least one year of data is necessary for both the effective training and the reliable validation of the algorithms. For this reason, all datasets used in this study cover the two full calendar years of 2019 and 2020. The temporal resolution of the PV power and NWP data is 15-min, which fits the typical requirements of the electricity markets in California [15], China [39], and Hungary.

The PV power production data, which is used as the predictand (i.e., the output variable), is obtained from the official measurements of 16 ground-mounted PV plants of the MVM Green Generation Ltd in Hungary. The plants have installed capacities between 0.5 and 20 MW, while their metadata, map, and further details are summarized in Ref. [10]. The numerical weather prediction (NWP) data, serving as the main predictors, include global horizontal irradiance (GHI), ambient temperature, and wind speed. These forecasts are generated by the daily 00 UTC runs of the operational AROME non-hydrostatic mesoscale model of the Hungarian Meteorological Services for a 24-48-h time horizon. A detailed description of the AROME model can be found in Ref. [40]. Day-ahead is often used as an equivalent of 24-h-ahead, however in energy sector applications, to obey, e.g., Hungarian regulations, a PV park owner must submit the complete DA schedule of the next day until 10:00 a.m. (local time). It means that even if the NWP process takes a few hours, the forecast used in this research is applicable for regional DA scheduling.

The pre-processing of all datasets includes the removal of nighttime data (zenith angle is higher than 90°) and the timesteps with zero PV power output, as daytime periods with absolutely no production indicate the shutdown of the plants. Otherwise, both the PV power and NWP

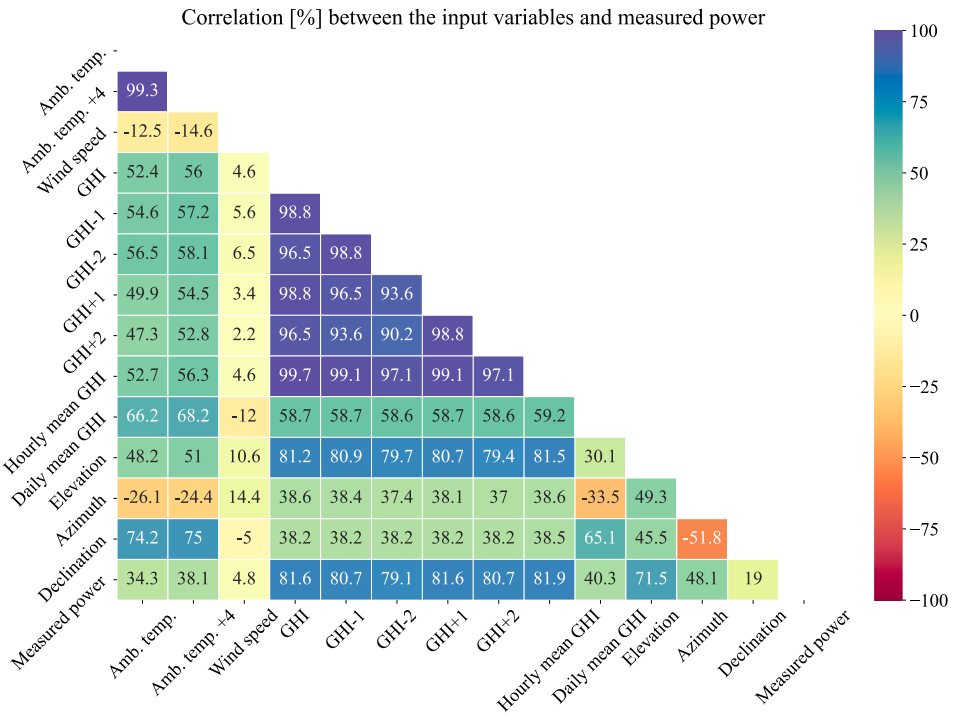


Fig. 1. Comparison of Pearson correlation coefficients between possible input data and measured power output at one location.

data are of good quality; therefore, no further quality control was required. As a result, only the daytime data with non-zero power production, covering 34000–35000 valid timesteps for each plant, are included in the training and testing of the models. The inputs and outputs are normalized to a mean of 0 and standard deviation of 1 prior to supplying them to the machine learning models to ensure a good convergence.

The three meteorological variables included in the NWP can be directly used as the inputs of the learning models, while further predictors can also be created by manipulating these data and simple physically inspired calculations. The declination, azimuth, and elevation angles, describing the position of the Sun in the sky, are calculated from the location of the plant and the date and time for each timesteps using the Solar Position Algorithm (SPA) [41]. In the present implementation, the data for each timestep is used as the individual training examples, which prevents the models from learning from the temporal variation of the solar irradiance. To utilize the extra information coming from the change of the weather around the timestep of interest, the irradiance from preceding and following timesteps can also be added to the inputs. For a similar reason, the moving average and the daily mean of the solar irradiance can also be used as further predictors.

As Fig. 1 shows, the input variables best correlated with the measured power production are the 15-min resolution and hourly mean GHI and elevation angle. These strong relations are followed by the

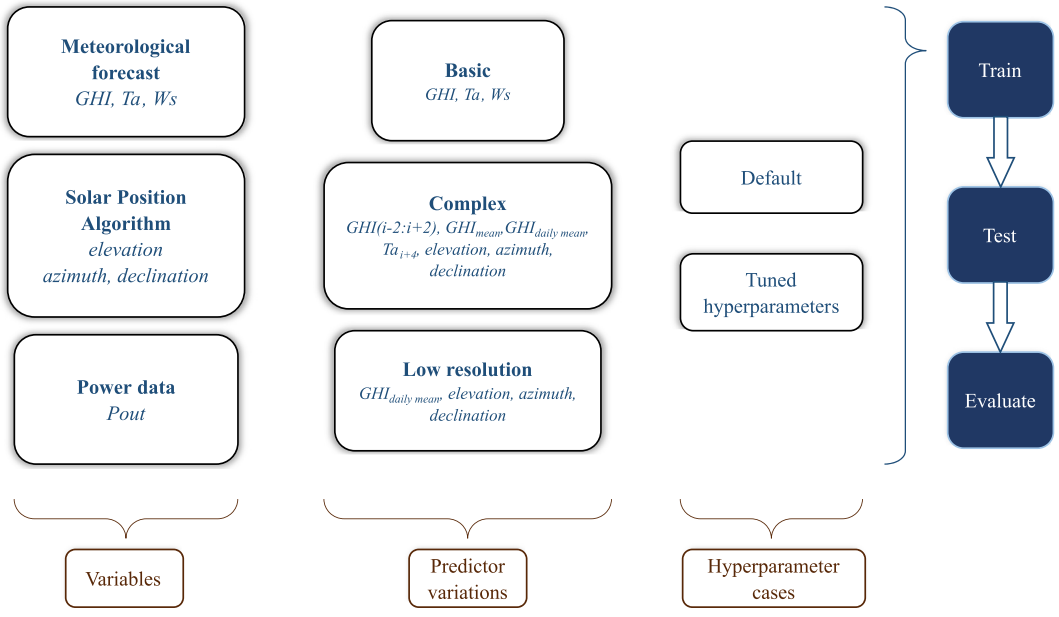


Fig. 2. Methodology of the compared predictor combinations in this study.

Table 2
Summary and brief descriptions of tested models and hyperparameters. All tested values of each hyperparameter are listed in brackets, with bold letters denoting the default values.

Family	Abbr.	Name in scikit-learn	Brief description	Tuned hyperparameters	Ref.
Linear models	LR	Linear Regression	calculates the predictand as a linear function of the predictors, coefficients fitted using the least squares method	normalize: [True, **False**]	
	Lasso	Lasso	linear regression with an L^1-norm penalty regularization (absolute values of coefficients are added to the loss function)	alpha: [0.01, 0.1, 0.5, 0.85, 0.95, **1**, 1.1, 1.3, 1.5, 10]	[46]
	Ridge	Ridge	linear regression with an L^2-norm penalty regularization (squared values of coefficients are added to the loss function)	alpha: [0.01, 0.1, 0.5, 0.85, 0.95, **1**, 1.1, 1.3, 1.5, 10]	[47]
	EN	ElasticNet	linear regression with both L^1 and L^2 regularizers	alpha: [10, 1.5, 1.3, 1.1, **1**, 0.95, 0.5, 0.1], l1_ratio: [0.3, **0.5**, 0.7]	[46]
	Lars	Least Angle Regression	linear model with a stepwise feature selection algorithm for high dimensional data based on finding the best absolute correlated explanatory variable at each step; when having multiple features with same correlation, continues the procedure in equiangular direction between those	n_nonzero_coefs: [1, 10, 100, 300, **500**, np.inf], eps: [10, 5, 3, 1, 0.7, 0.5, 0.1] **np.finfo(float).eps**	[48]
	OMP	Orthogonal Matching Pursuit	adding forward selective OMP algorithm to linear model fitting approximation of optimum solution vector, which at each step includes the highest correlated residual recomputed by an orthogonal projection on previous data	n_nonzero_coefs: [**None**] + list(range(1, len(inputs)), normalize: [**True**, False]	[49]
	BR	Bayesian Ridge	creates a probabilistic model with estimated maximum log marginal likelihood regularization parameters included, similar to basic Ridge	alpha_1: [**0.000001**, 0.00001, 0.0001, 0.001], alpha_2: [**0.000001**, 0.0001, 0.001, 0.01, 0.1], lambda_1: [**0.000001**, 0.0001, 0.001, 0.01, 0.1], lambda_2: [**0.000001**, 0.00001, 0.001, 0.01]	[50]
	ARD	ARD Regression	bayesian automatic relevance detection weight fitting (Gaussian distribution) to regression model with the purpose of evidence maximization	alpha_1: [**0.000001**, 0.0001], alpha_2: [**0.000001**, 0.1, 0.5], lambda_1: [**0.000001**, 0.0001, 0.1], lambda_2: [**0.000001**, 0.0001, 0.001, 0.1], threshold_lambda: [5000, **10000**]	[51]
	PAR	Passive Aggressive Regressor	large scale learning algorithm without learning rate, including regularization parameter C	C: [**1.0**, 1.2, 1.5, 2, 3, 4], epsilon: [0.001, 0.01, 0.05, **0.1**, 0.8], average: [**False**, True]	[52]
	RANSAC	RANSAC Regressor	RANdom SAmple Consensus non-deterministic algorithm providing output with a certain probability defined by max_trials parameter	max_skips: [5, 10, 15, 20, **np.inf**], max_trials: [5, 10, 15, 20, **100**], min_samples: [0.8, 0.85, 0.9, **None**], stop_n_inliers: [1, 3, 5, **np.inf**], stop_probability: [0.05, 0.1, **0.99**]	[53]
	TR	TheilSen Regressor	least square based solution for extremely large datasets, with parameters to set limit to robustness	max_subpopulation: [**1000**, 5000, 10000]	[54]
	Huber	Huber Regressor	linear regression-based model with epsilon and alpha parameters to remove outliers	alpha: [10, 1, 0.1, 0.01, 0.001, **0.0001**], epsilon: [1.1, 1.4, 1.8, 2.2]	[55]
Kernel Ridge	KR	Kernel Ridge	Kernel Ridge is a ridge regression in the kernel space, meaning that the input is non-linear transformed into higher dimension feature space, where the calculations are carried out (kernel trick), providing an upper limit to complexity	kernel: [‘polynomial’, ‘rbf’, **‘linear’**]	[56]
Support Vector Machines	SVM	Support Vector Regression	Support Vector Machine, non-linear transforms inputs into high dimensional kernel space (kernel trick), and fits the optimal hyperplane to get the best possible linear solution	kernel: [**’rbf’**], epsilon: [0.5, 0.25, 0.23, 0.2, 0.15, 0.12, **0.1**, 0.08], C: [0.001, 0.01, 0.1, 0.5, 0.9, **1**, 1.15]	[57]
Neural Networks	MLP	MLP Regressor	MultiLayer Perceptron regressor, the basic type of feedforward neural networks, trained using backpropagation technique using the partial derivatives of the loss function to update parameters	hidden_layer_sizes: [(5,), (10,), (50,), (5,5), (10,10), (5,5,5), **(100,)**], alpha: [**0.0001**, 0.001, 0.01, 0.1, 1]	[58]
Nearest Neighbors	KNN	KNeighbors Regressor	k Nearest Neighbors, the output is calculated as the average of the *k* nearest point in the input variable space	n_neighbors: [**5**, 10, 15, 20, 30, 50, 70, 100, 200, 300, 500]	[59]
Decision Trees	DT	Decision Tree Regressor	uses a simple structure that decides through a series of tests, where each internal node in the tree satisfies the examination of the value of a property, and the branches exiting the node are indicated by the possible outputs of the test	max_depth: [**None**, 5, 8], min_samples_leaf: [0.01, 0.001, **1**], max_features: [’auto’, ‘log2’, ‘sqrt’, **None**]	[60, 61]
Ensemble methods	RF	Random Forest Regressor	ensemble method, fits definite number of decision trees on parts of the data, and uses their average to improve	n_estimators: [50, **100**, 250, 300], max_depth: [**None**, 5, 8], min_samples_leaf: [**100**, 10, **1**, 0.01], max_features: [**’auto’**, ‘sqrt’]	[61]

(*continued on next page*)

Table 2 (*continued*)

Family	Abbr.	Name in scikit-learn	Brief description	Tuned hyperparameters	Ref.
	ET	Extra Trees Regressor	similar ensemble to random forest but with increased randomness to reduce variance, at expense of bias (averages randomized decision trees fitted on sub-samples)	n_estimators: [50, **100**, 250, 300], max_depth: [**None**, 5, 8], min_samples_leaf: [100, 10, **1**, 0.01], max_features: [**'auto'**, 'sqrt']	[62]
	ABR	AdaBoost Regressor	Adaptive Boosting, ensemble method implementing AdaBoost.R2 algorithm that fits sequence of decision trees and modifies the weights of samples after each iteration to repeat procedure	n_estimators: [**50**, 100, 250], learning_rate: [**1**, 0.1, 0.01], loss: [**'linear'**, 'square', 'exponential']	[63]
	GBR	Gradient Boosting Regressor	generalized boosting ensemble algorithm with adjustable loss function	alpha: [0.85, **0.9**, 0.95], n_estimators: [25, 50, 70, **100**, 120], loss: ['ls', 'huber', **'squared_error'**]	[64]
	XGBoost	XGB Regressor	Extreme Gradient Boosting, substantive library of boosted trees developed to push the extreme of the computation limits for scalability, portability, and accuracy	learning_rate: [0.01, 0.05, 0.1, **0.3**], max_depth: [3, 4, 5, **6**], min_child_weight: [**1**, 2, 3, 4], n_estimators: [50, **100**, 200], reg_lambda: [0.1, 0.4, **1**]	[65]
	LGBM	LGBM Regressor	Light Gradient boosting Machine, a boosting package that uses histogram-based algorithm to speed up training and reduce memory usage	num_leaves: [3, 6,8, 10, 15, 25, **31**, 50, 100], max_depth: [**-1**, 3, 5, 6, 8, 10, 15, 25, 50]	[66]
	CatBoost	CatBoost Regressor	gradient boosting toolkit implementing ordered boosting and an innovative algorithm for processing categorical features	depth: [3, 4, 5, **6**, 8], l2_leaf_reg: [**3**, 10, 50, 100, 200, 500, 1000]	[67]

azimuth angle, daily mean GHI, and T_a variations. In the case of ambient temperature, values shifted by 1 h (T_a+*4*) proved to have higher explanatory power. The wind speed does not correlate with the measured power output at all, which supports the observations that may even worsen the forecast accuracy, which is in line with the findings of previous studies [10]. Such a low correlation between wind speed and PV power has already been shown in a recent review [42].

The algorithms have been tested for the following three different sets of predictors (input data cases):

- *Basic:* GHI, ambient temperature, and wind speed forecasts. These inputs are qualified as basic since these are the most commonly used predictors in the previously described research papers.
- *Complex:* GHI from the actual, the preceding two, and the following two timesteps, hourly and daily mean GHI, ambient temperature shifted by 1 h, and the elevation, azimuth, and declination angles. This set of predictors was found to be the most effective during the preliminary assessments of the present study, and they can be easily obtained from the basic inputs (and the time and location) without any additional data requirement.
- *Low-resolution:* Daily mean GHI, supplemented by the modeled 15-min elevation, azimuth, and declination angles. This input set can be used without purchasing any high-resolution (15-min) NWP forecast, which is useful if only a low budget is available for the PV power forecasting.

Fig. 2 shows the methodology explained in the past few paragraphs. In the first column, the input dataset is categorized by source, which is followed by the compared predictor combinations summarizing the variables used in each case. As the third column indicates, these parameter combinations are evaluated with and without hyper-parameter tuning.

2.2. Machine learning models

Machine learning (ML) is a powerful tool for engineering applications, especially when physically inspired modeling is ineffective due to the complexity of the unknown properties of the described phenomena. As physical modeling of PV plants requires a model chain of multiple steps and the knowledge of the plant design data [10], ML is a simple and widely used alternative for PV power calculations [43].

The most commonly used branch of ML is supervised learning, where the training data includes the values of both the predictors and the predictand [44]. The purpose of supervised learning is to find the best connection between the inputs and outputs. Generally, machine learning problems can be divided into two main categories: classification, when the outputs are categorical variables with discrete values, and regression, when the outputs are continuous. PV power forecasting is a typical regression problem.

The application of ML starts with creating an ML model and fitting its parameters to the training dataset. This training procedure is accomplished through an optimization routine that aims to find the parameters that can describe the relationship between the inputs and outputs with the possible lowest errors. The goodness of the trained model can only be reliably evaluated on previously unseen test data. For this reason, the available historical data must be split into training and test datasets. In this study, a two-fold cross-validation method is used, where the accuracy of the model fitted for the 2019 data is evaluated for 2020, and vice versa. This cross-validation enables to include the whole dataset in the evaluation and reduces the uncertainty resulting from any arbitrary train-test split.

On a higher level of abstraction, there are numerous ML models that can be used with this exact same general procedure. However, the underlying logic and mathematical structure of these models are different; therefore, they all have different performances for a particular type of regression problem. In this study, the 24 different models presented in

Table 3
Summary of the five metrics and the runtime, averaged for the 16 tested PV plant, for all 24 machine learning models and the six input and tuning cases. Green and red colors stand for the better and worse than median values, while bold numbers indicate the row-wise best results.

	Input	Hyperp.	Linear models													Kernel Ridge	Support Vector Machines	Neural Network	Neighbors	Decision Tree	Ensembles						
			LR	Lasso	Ridge	EN	Lars	OMP	BR	ARD	PAR	RAN-SAC	TR	Huber		KR	SVM	MLP	KNN	DT	RF	ET	ABR	GBR	XG-Boost	LGBM	Cat-Boost
nRMSE	Basic	default	52.6	94.0	52.6	78.1	52.6	53.3	52.6	52.6	81.2	54.0	52.8	53.0		52.6	53.2	52.1	59.6	72.9	56.2	56.3	56.5	**52.1**	55.6	53.6	54.2
		tuned	52.6	52.6	52.6	53.4	52.6	52.5	52.6	52.5	52.7	**53.8**	52.8	52.6		**51.5**	52.5	51.5	52.8	52.7	52.0	51.7	53.2	51.7	51.8	52.4	51.7
	Complex	default	49.4	94.0	49.4	72.6	58.4	52.8	49.4	49.4	77.1	56.7	49.9	50.1		49.4	46.7	46.7	53.1	63.9	48.7	47.7	51.8	**46.7**	50.6	48.4	48.8
		tuned	49.4	49.7	49.4	49.8	49.5	49.4	49.4	49.4	49.4	50.3	49.9	49.4		**45.3**	46.6	45.4	46.4	48.2	46.1	45.9	49.7	46.5	46.4	46.7	46.1
	Low resolution	default	52.1	94.0	52.1	84.8	52.1	65.7	52.1	52.1	79.9	56.1	51.2	52.2		52.1	47.2	**46.2**	54.2	64.6	52.7	51.4	55.5	48.0	52.2	49.9	50.4
		tuned	52.1	52.6	52.1	55.1	52.6	63.2	52.1	52.1	51.5	55.3	51.4	52.1		**46.0**	46.6	46.2	47.1	51.3	49.1	47.8	57.3	48.2	48.3	48.8	47.4
nMAE	Basic	default	37.4	82.7	37.4	68.8	37.4	38.2	37.4	37.4	59.9	36.3	36.2	35.8		37.4	**35.1**	36.5	40.9	48.7	39.0	39.1	44.9	36.6	38.5	37.2	37.6
		tuned	37.4	37.7	37.5	40.4	37.4	37.4	37.4	37.4	37.5	36.4	36.2	36.8		36.2	37.1	36.3	38.0	36.9	36.4	36.5	38.9	**36.0**	36.8	36.7	36.6
	Complex	default	35.3	82.7	35.3	63.7	40.9	37.8	35.3	35.3	56.0	37.9	34.3	34.3		35.3	**27.9**	30.8	32.7	38.8	31.0	30.7	39.5	30.9	32.6	30.9	31.5
		tuned	35.3	35.7	35.3	35.9	35.4	35.3	35.3	35.3	35.0	34.2	34.3	34.9		**29.6**	30.8	29.6	30.4	31.8	30.2	30.2	36.4	30.8	30.8	30.9	30.5
	Low resolution	default	39.8	82.7	39.8	74.8	39.8	49.8	39.8	39.8	58.8	39.5	38.2	38.5		39.8	**29.2**	30.4	34.3	39.9	33.5	33.1	43.2	32.8	34.0	32.4	32.9
		tuned	39.8	40.3	39.8	43.4	40.1	48.0	39.8	39.8	38.7	39.6	38.3	39.6		31.0	**30.3**	31.0	32.2	34.9	33.4	33.0	42.5	33.1	33.7	34.8	32.5
Corr	Basic	default	82.9	-1.1	82.9	82.1	82.9	82.3	82.9	82.9	61.3	82.5	82.9	82.9		82.9	83.0	**83.3**	78.1	69.6	80.3	80.2	81.1	83.2	80.8	82.2	81.8
		tuned	82.9	82.9	82.9	82.6	82.9	82.9	82.9	82.9	82.9	82.4	82.9	82.9		**83.7**	83.0	83.6	82.8	82.8	83.3	83.5	82.5	83.5	83.4	83.0	83.5
	Complex	default	85.1	-1.1	85.1	82.6	78.7	82.7	85.1	85.1	65.3	81.0	84.9	84.7		85.1	**87.2**	86.8	83.2	76.7	85.6	86.2	84.5	86.8	84.5	85.8	85.5
		tuned	85.1	84.9	85.1	84.8	85.0	85.1	85.1	85.1	85.3	84.6	84.9	85.0		**87.6**	86.9	87.6	87.0	85.9	87.2	87.3	85.2	86.9	86.9	86.8	87.1
	Low resolution	default	83.6	-1.1	83.6	70.9	83.6	71.5	83.6	83.6	66.6	81.0	84.0	83.2		83.6	86.8	**87.1**	82.3	76.1	83.1	83.9	81.7	86.0	83.5	84.9	84.6
		tuned	83.6	83.3	83.6	82.1	83.2	73.9	83.6	83.6	84.1	81.1	83.9	83.5		**87.2**	86.9	87.1	86.6	83.8	85.3	86.2	79.2	85.8	85.8	85.5	86.4
Skill	Basic	default	35.7	-15.0	35.7	4.4	35.7	34.8	35.7	35.7	0.6	34.0	35.4	35.2		35.7	34.9	36.3	27.1	10.8	31.3	31.1	30.8	**36.3**	32.0	34.5	33.7
		tuned	35.7	35.7	35.7	34.7	35.7	35.7	35.7	35.7	35.6	34.2	35.4	35.7		**37.1**	35.7	37.0	35.5	35.6	36.5	36.8	34.9	36.7	36.7	35.9	36.8
	Complex	default	39.7	-15.0	39.7	11.1	28.7	35.4	39.7	39.7	5.8	30.7	39.0	38.7		39.7	42.9	42.9	35.1	21.8	40.4	41.6	36.7	**42.9**	38.1	40.8	40.3
		tuned	39.7	39.2	39.7	39.1	39.5	39.7	39.7	39.7	39.6	38.5	39.0	39.6		**44.6**	43.0	44.5	43.3	41.1	43.7	43.9	39.2	43.1	43.2	42.9	43.6
	Low resolution	default	36.3	-15.0	36.3	-3.8	36.3	19.7	36.3	36.3	2.3	31.4	37.4	36.1		36.3	42.3	**43.5**	33.7	21.0	35.5	37.1	32.1	41.3	36.1	39.0	38.4
		tuned	36.3	35.7	36.3	32.7	35.7	22.7	36.3	36.3	37.0	32.4	37.2	36.4		**43.8**	43.0	43.5	42.5	37.2	39.9	41.6	29.9	41.1	41.0	40.3	42.1
nMBE	Basic	default	0.16	**0.00**	0.16	0.00	0.16	-0.01	0.15	0.15	-1.16	-3.97	-1.78	-2.98		0.16	-0.73	-0.20	-0.98	-1.04	-0.60	-0.53	4.96	0.03	-0.40	-0.25	-0.28
		tuned	0.16	0.12	0.15	0.09	0.14	0.16	0.15	0.16	3.14	-3.63	-2.03	-0.18		0.14	2.71	0.03	-1.22	-0.03	0.00	-0.10	0.68	-0.39	0.26	**0.00**	0.10
	Complex	default	1.14	**0.00**	1.14	0.00	2.61	-0.02	1.13	1.13	11.38	-0.46	-0.88	-0.90		1.14	2.65	0.42	0.58	-0.02	-0.04	-0.11	-1.34	0.08	-0.01	0.05	0.03
		tuned	1.14	0.49	1.13	0.42	0.97	1.14	1.13	1.12	5.89	-0.50	-1.04	1.25		0.37	3.04	0.09	0.06	0.14	**-0.02**	0.03	-0.30	-0.15	0.29	0.07	0.09
	Low resolution	default	2.90	**0.00**	2.90	0.00	2.90	0.19	2.90	2.90	17.27	3.48	4.24	1.97		2.90	2.19	0.67	0.47	-0.58	-0.46	-0.47	-0.18	0.76	0.50	0.47	0.66
		tuned	2.90	2.47	2.89	1.09	2.68	0.71	2.90	2.90	8.44	-3.42	4.04	3.17		0.81	1.64	0.31	1.61	0.19	0.21	**0.11**	0.73	0.75	0.76	0.79	0.39
Time	[s]	default	0.2	0.1	0.1	0.2	0.2	0.1	0.2	0.5	0.4	2.0	130.5	4.4		1091.8	700.9	321.8	6.8	6.0	24.9	11.6	20.3	132.	18.7	3.6	120.0
		tuned	0.2	1.8	0.1	0.5	0.2	0.1	0.2	0.3	0.5	0.8	98.1	3.9		1044.6	597.1	73.5	23.9	2.5	15.4	8.7	278.5	83.7	9.4	1.6	99.0

Table 2 are compared for the irradiance-to-power conversion step of PV power forecasting. The first 21 of these models are taken from the Python *scikit-learn* library [45], while *XGBoost*, *LGBM* and *CatBoost* are from independent packages. As the detailed working principles of these models have already been presented in a high number of previous studies, the authors see no added value in reiterating it in this paper. A summary of each model is given in Table 2, while interested readers can find their more detailed description in the supplemented references.

2.3. Hyperparameter tuning

Machine learning models have parameters and hyperparameters [68]. The parameters describe the connection between the inputs and outputs, and they are set during the training phase. The hyperparameters specify the structure of the model and control the training process; thus, they have to be chosen in the model initialization stage. The selection of the models can affect both the convergence of the training and the accuracy of the trained model. As the best hyperparameter values depend on the problem of interest, the highest possible accuracy can only be achieved by optimizing the main hyperparameters, which process is also called the tuning of the model. Most machine learning libraries, including the *scikit-learn* used in this study, have default sets of hyperparameters for each ML algorithm, which enables the use of all implemented models without digging deeper into their detailed working principle and the effect of the hyperparameters. Even though these default hyperparameter sets offer a decent accuracy in most applications, the full potential of an ML model can only be unlocked by a tuning process.

The tuning is performed by training the evaluating the models for different hyperparameter combinations, using the two-fold cross-validation technique introduced in Section 2.3, and choosing the set of hyperparameters that resulted in the lowest errors. Generally, the two main approaches for tuning are grid search and random search [69]. In grid search, all possible combinations of the pre-defined hyperparameter values are evaluated, which can lead to an unacceptable long calculation time if the number of combinations is high, i.e., there are several hyperparameters with many different tested values for each of them. This can be avoided by lowering the resolution of the search space by limiting the tuning for only a few different values for each hyperparameter. Alternatively, a random search [70] can be used, which only tests a pre-defined number (typically a few tens) combinations, making it faster even for high-resolution hyperparameter grids; however, it does not guarantee that the actual best combination is found. A common feature of the two principles is that they do not use feedback from the previous results of tested combinations. To further increase accuracy and simultaneously reduce resource requirements and runtime, other sophisticated strategies, like Bayesian Optimization [70] or Successive Halving [71], can be applied using either *scikit-learn*, or other libraries specifically designed for this purpose, e.g., *Sherpa* [72]. In order to enhance the consistency and reproducibility and reduce the randomness of the tuning process, grid search is used in this study. The tuned hyperparameters for each model, along with their tested values, are included in Table 2. For further information about recent developments and future directions of hyperparameter tuning techniques, the work of Khalid et al. [73] is recommended.

2.4. Verification metrics

The aim of verification is to measure the goodness of the forecasts. As shown in Table 1, many different metrics are used for this without a commonly accepted agreement, making it hard to compare the results of different studies. The need for a unified and reliable verification method is manifested in the recent paper of Yang et al. [7], which summarizes the main guidelines and recommendations for the verification of deterministic solar forecasts. One of the main concepts emphasized in Ref. [7] is consistency, which means that the forecasts should always be

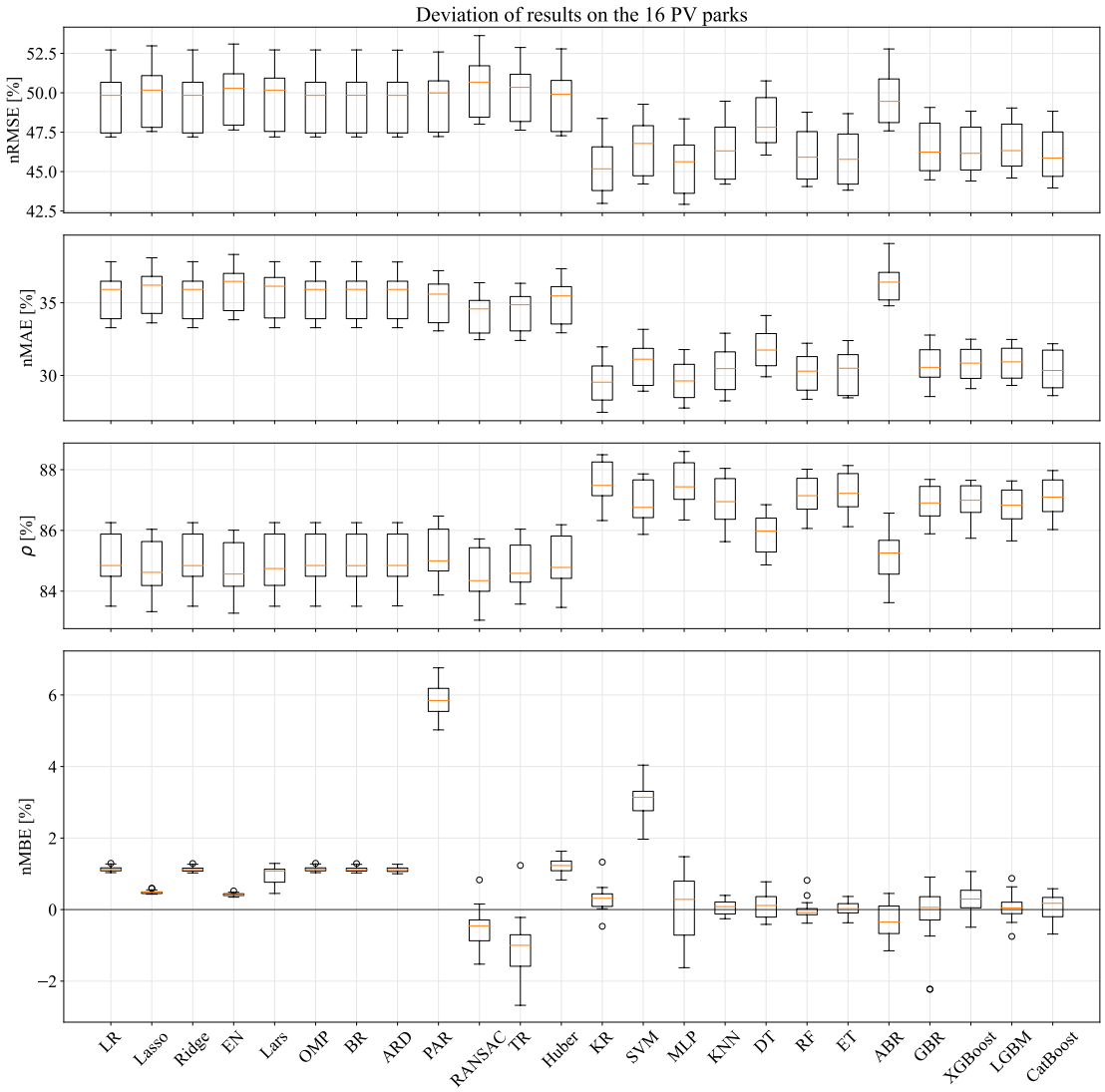


Fig. 3. Boxplots of nRMSE, nMAE, correlation coefficient, and nMBE of the power forecasts calculated by the different models for the 16 PV parks for the complex tuned case.

evaluated using the same metrics as they were fitted or optimized for [74]. The training process of most ML methods minimizes the mean square error (MSE) of the outputs; therefore, root mean square error (RMSE) is used as the main verification metric. Furthermore, RMSE has often been considered the best indicator of the value of forecasts in utility applications, as it punishes high errors disproportionally more than smaller ones [17]. Still, RMSE alone is not enough to completely describe the quality of forecasts; therefore, it should be supplemented by other metrics.

The mean bias error (MBE) indicates whether the forecasts under- or overpredict on average. Pearson's correlation coefficient (ρ) reflects the association between forecasts and observations and the potential skill of the forecasts regardless of their calibration, i.e., their bias and variance [75]. The RMSE skill score (s) shows the proportion of the RMSE improvement over a reference method, which provides some degree of inter-comparability for locations with different solar irradiance variability [13]. The hereby applied and overall most commonly used reference method for day-ahead forecasting is the 48-h persistence, which uses the measured production of the second last day (which is the last fully available day when the day-ahead forecasts are issued) as the forecasts for the day of interest [14]. Finally, mean absolute error (MAE) is also calculated, as the total imbalance penalties are proportional to the absolute difference of power forecasts and observations in most markets [76]. Even though the nMAE can be directly interpreted as the balancing energy need for a unit of electricity production, the balancing energy prices may also tend to be proportional to the errors as for a higher imbalance, more expensive resources are required from the merit order list of reserve power plants [77], which is then better reflected by the RMSE. As long there is no consensus regarding the error metric that best represents the financial value of PV forecasts, both of them are advised to be included in forecast verifications.

According to the RMSE decomposition shown in Refs. [12,75], the lowest RMSE requires zero bias and the highest possible correlation coefficient, which means that optimizing the models for RMSE during training also aims to reach a low MBE and high correlation. In other words, these metrics are consistent with the target (i.e., minimizing RMSE) of the training process, and so is the skill score, which is just a linear transformation of the RMSE. However, MAE and RMSE are two conflicting error metrics [10], as the absolute error is minimized by the median, while square error is minimized by the mean of the underlying predictive densities [78]. Therefore, the MAE should only be treated as additional information for a better understating of the properties of the produced forecasts instead of using it as a main evaluation criterion during the interpretation of the results.

The RMSE, MAE, and MBE are normalized by the average daytime PV power for the comparability of PV plants with different nominal

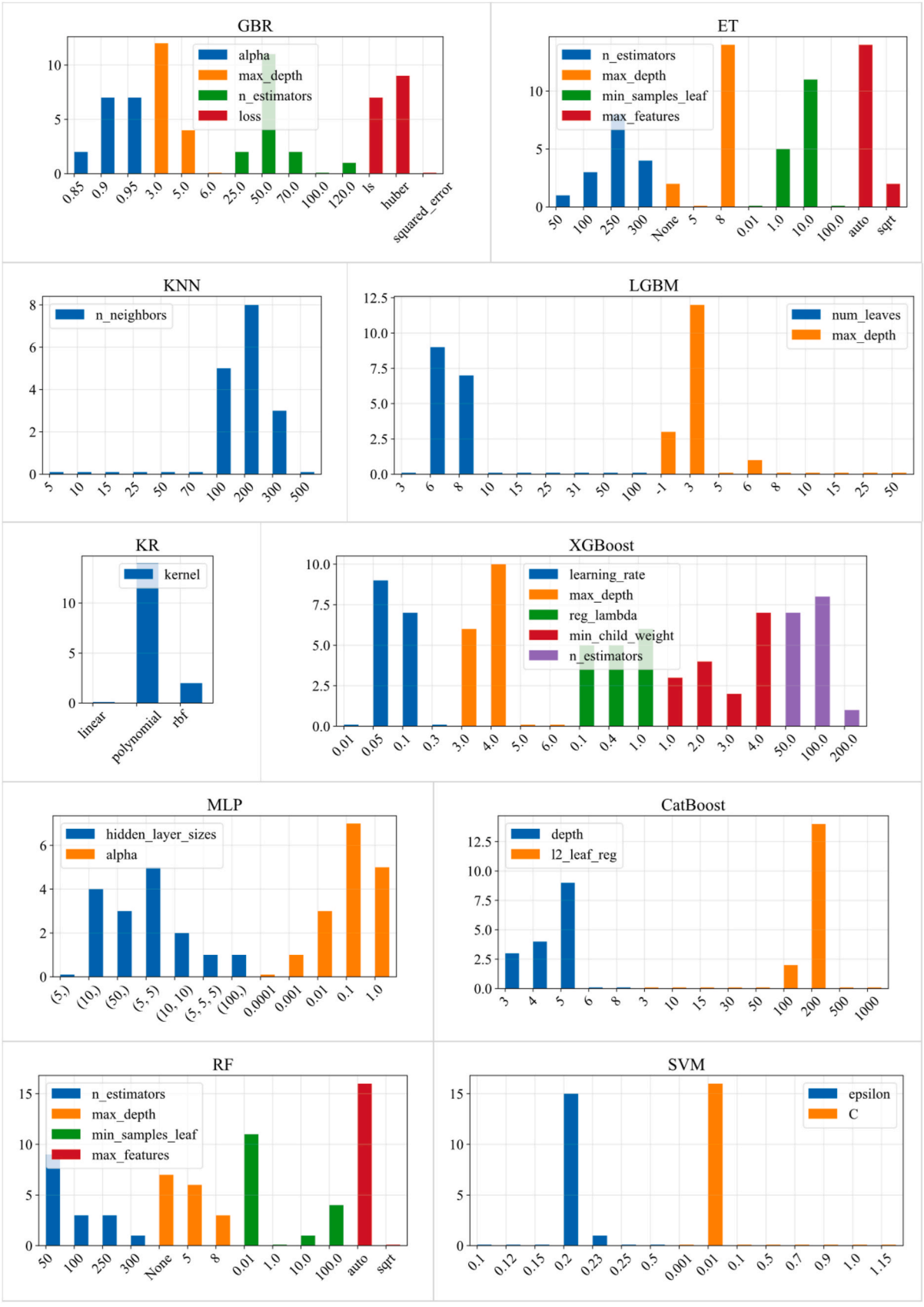


Fig. 4. Frequency of used hyperparameters in the best combinations for the 16 PV parks for the nine most promising models in complex tuned case.

power and energy production. The metrics are calculated using the following formulas,

$$\mathrm{nRMSE} = \frac{\sqrt{\frac{1}{N}\sum_{i=1}^{N}(f_i - P_i)^2}}{\frac{1}{N}\sum_{i=1}^{N}P_i} \quad (1)$$

$$\mathrm{nMAE} = \frac{\sum_{i=1}^{N}|f_i - P_i|}{\sum_{i=1}^{N}P_i} \quad (2)$$

$$\mathrm{nMBE} = \frac{\sum_{i=1}^{N}(f_i - P_i)}{\sum_{i=1}^{N}P_i} \quad (3)$$

$$\rho = \frac{\mathrm{cov}(P_i, yf_i)}{\sigma_{P_i}\sigma_{f_i}} \quad (4)$$

$$s = 1 - \frac{\mathrm{RMSE}_{\mathrm{forecast}}}{\mathrm{RMSE}_{\mathrm{persistence}}} \quad (5)$$

where i is the number of the given timestep, f_i is the predicted and P_i is the measured output power, cov(..) is the covariance, σ is the standard deviation, and N is the length of the daytime dataset. As the training, testing, and tuning of the models is performed separately for all 16 PV plants, the error metrics are also calculated separately, and they are simply averaged for the overall evaluations. Finally, it is important to note that simple post-processing was applied to the power prediction before their verification by clipping their values into the range between zero and the maximum measured power output of the training dataset. This modification reduced the effects of overfitting and softened the difference between good and weak methods.

Finally, the runtime is used as a sixth indicator, which is interpreted as the sum of the training and prediction time, to represent the computation requirements of the model. The process was conducted on a personal computer with 16 GB RAM and an AMD Ryzen 9 3900x CPU with 12 cores and 24 threads.

3. Results and discussion

This section presents and discusses the main results of the study. The five metrics selected for the verification, averaged for the 16 examined PV plants, are presented for all 24 ML models and the six input and tuning cases in Table 3. The detailed interpretation and discussion of these main results, covering the comparison of the ML models and the effects of the input data and the hyperparameter tuning, is presented in the following subsections. Finally, a visual evaluation of the most promising power forecasts is also included to supplement the metric-based conclusions.

3.1. Comparison of the machine learning models

According to Table 3, most of the nonlinear models outperform the linear ones in terms of all five metrics in all cases, which is, in itself, not surprising due to the nonlinear relationship between the GHI and the PV power. Observing only the linear models, most of the more advanced models are less accurate than the simple LR; therefore, there seems to be no significant potential in the advanced linear models in this kind of application. Moreover, some of them (Lasso, EN, OMP, PAR, and RANSAC) are prone to severe underfitting, especially without hyperparameter tuning, which largely increases their error compared to the LR. The only advantage of the linear models is their significantly shorter training time than the average of the nonlinear ones, which can make them favorable if fast training is crucial.

The performance of the nonlinear models varies in a wide range, which means that nonlinearity in itself is not enough for good accuracy. For example, ABR, despite its long training time, is less accurate than even the LR in all cases. On the other end, the overall most accurate models, with head-to-head performance, are the KR and the MLP. The good performance of the MLP has already been shown is previous studies [9,31], but KR is not commonly used in the field of solar forecasting. However, the practical usability of KR is largely limited by its exceptionally long computational demand, as it takes more than one order of magnitude more time to train KR than the MLP, which has an around average training time among the nonlinear models. SVM is also notable for its low MAE in some of the cases, even though it takes a long time to train it. The fastest nonlinear model is the DT; however, its accuracy falls behind most of its competitors. KNN, XGBoost, and LGBM also have a short training time, but it is paired with only an average performance.

Theoretically, there is no way to find a universally optimal tradeoff between the accuracy and the training time, as it always depends on the benefits of the reduced errors and the cost of the computational resource. In PV forecasting applications, the added financial value of more accurate forecasts is typically considerably higher than the costs associated with the larger computation requirements. However, in the field of automated machine learning, it is a common way to use a time budget for the training and tuning of the model during benchmarking [79]. Overall, low errors are considered much more important in practical forecasting applications than training time; however, it is advised to choose the faster model if multiple models have almost identical performance.

The ranking of the different models and cases is fully consistent in terms of three of the five metrics, namely the RMSE, correlation coefficient, and skill score. The judgments based on the MAE are also similar; however, there are several exceptions, among which the most notable is the SVM, which excels in nMAE but not in RMSE. The MBE does not correlate clearly with the other metrics. The models with low RMSE typically have a low bias, but it is not true in reverse, i.e., a low MBE does not imply a good performance in terms of the other metrics.

While the metrics of Table 3 are averaged for all 16 PV plants, Fig. 3 shows the variability of performance between the plants by boxplots. These plots are limited to the complex tuned case, as it has the overall best accuracy; thus, this case is the most relevant in practice. The differences between the observed plants can be traced back to the regional climate and topography differences already observed in Ref. [10]. The spread of the nRMSE and nMAE does not depend significantly on the ML model, which means that all models can be considered reliable as their have a uniform performance, i.e., their additional irradiance-to-power conversion error does not depend considerably on the PV plants. The spread of the correlation coefficient is slightly smaller for the nonlinear models than the linear ones, which means that the more accurate models also have an equalizing effect on this metric.

The most informative part of Fig. 3 is related to the MBE, as it reveals that the close to zero nMBE values of Table 3 only mean that there is no general tendency for under- or overestimation but do not guarantee a low bias in all PV plants. A good example of this is MLP, which seems to be almost bias-free in Table 3, while it has an even higher than 2% nMBE in one of the PV plants. Even though MLP is one of the best models, it has the highest nMBE range, which suggests that the bias of MLP is quite sensitive to the regional and yearly weather differences. Finally, it is important to note that the post-processing applied on the power forecast, i.e., clipping the negative and the overly high values, can also introduce bias even if the raw output of the models is bias-free. This tendency is the most visible for the linear models, where replacing the negative values with zero increases the mean of the forecasts, which results in a positive bias.

3.2. Effect of the input data

Comparing the three input scenarios in Table 3 shows that the input selection has a higher effect on the error metrics than the model selection. Even the worst tuned model with the complex set of predictors has a lower RMSE than the best model with the basic inputs. Using the best

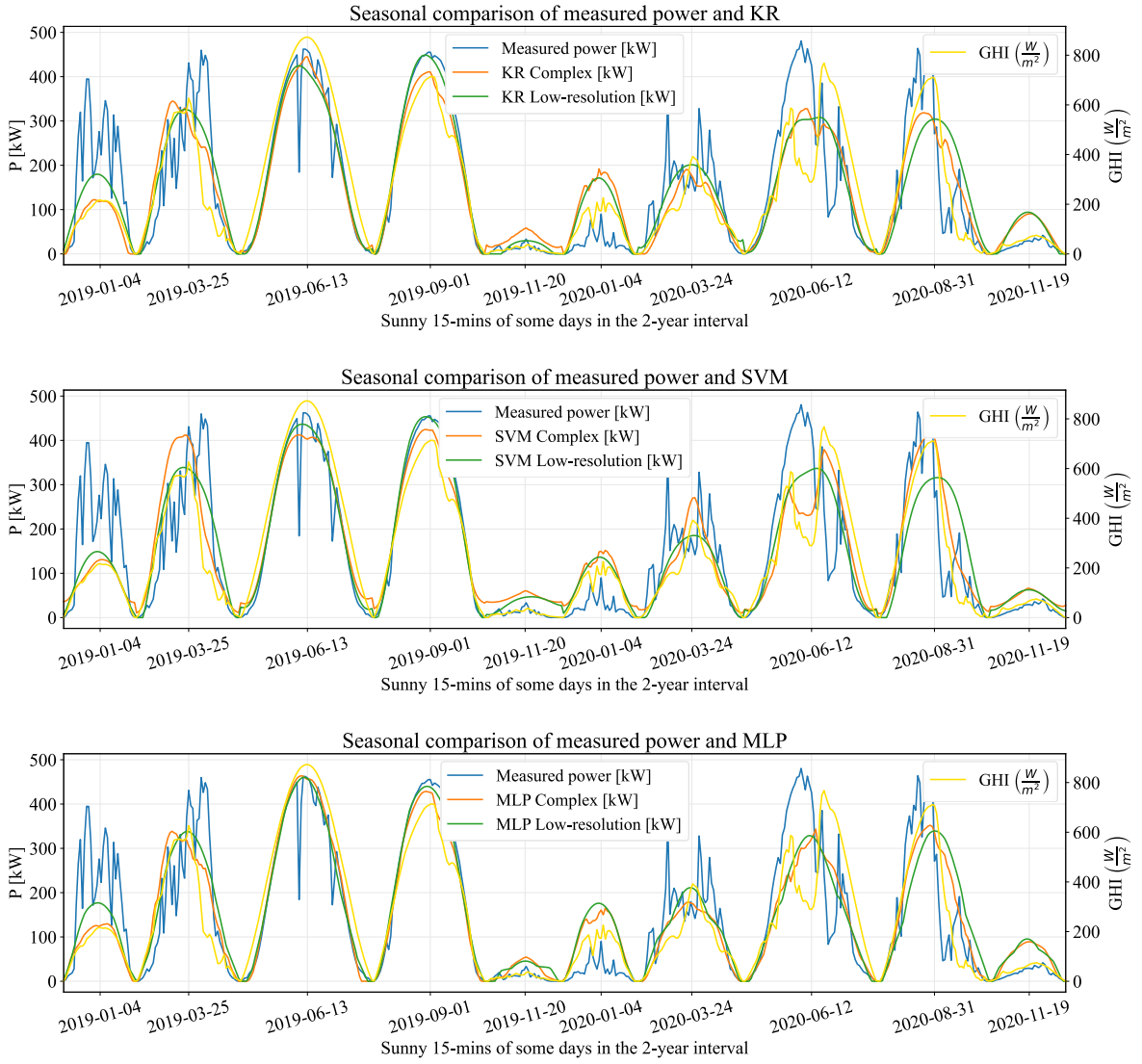


Fig. 5. Comparison of the measured power and the power forecast curves calculated by the KR, MLP, and SVM models in complex and low-resolution input cases on randomly chosen days of the 2-year interval.

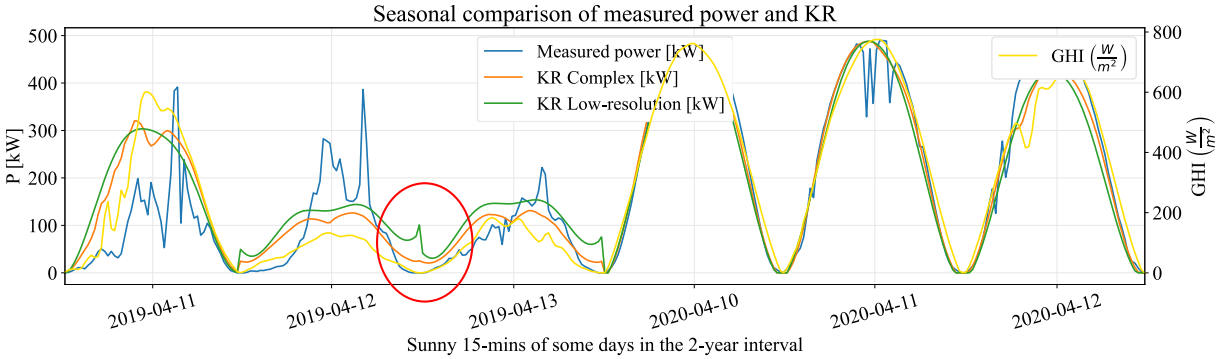


Fig. 6. Importance of visualization for error identification, shown on the KR model as an example, with a red circle indicating an unrealistic overestimation around sunrise and sunset.

Table 4
Overall evaluation and recommendation of the 24 compared machine learning models for solar irradiance to PV power conversion.

Family	ID	Accuracy	Hyperparameter sensitivity	Notes and recommendations
Linear models	LR	low	low	Fast and easily applicable, recommended to use it as a benchmark method
	Ridge	low	low	Same results as LR
	EN	low	high	Same results as LR, selection of penalty parameter is crucial
	Lasso	low	high	Same results as LR, selection of penalty parameter is crucial
	Lars	low	medium	Worse than basic LR
	OMP	low	medium	Worse than basic LR
	BR	low	low	Same results as LR
	ARD	low	low	Same results as LR
	PAR	low	high	Worse than basic LR
	RANSAC	low	high	Least accurate model even after tuning
	TR	low	low	Long training time despite being a linear model
	Huber	low	medium	Worse than basic LR
Kernel ridge regression	KR	high	medium/high	Best accuracy with polynomial kernel, but very slow with high memory usage, inhibiting parallel running at e.g., GridSearchCV
Support Vector Machines	SVM	high	low/medium	Very slow, but accurate nMAE performance with default parameters
Nearest Neighbors	KNN	high	high	Fast but only average performance, should be used with a relatively large k value (100–300)
Neural Networks	MLP	high	low/medium	Second best accuracy, long but still reasonable training time, high customization potential
Decision Trees	DT	low	high	Worse than most of the other nonlinear models, not recommended
Ensemble methods	RF	medium	high	From ensemble models use ET instead
	ET	high	high	More accurate than other similar methods
	ABR	low	high	The only nonlinear model that performs worse than LR
	GBR	medium	low	Acceptable default performance and easily applicable for quantile forecast, but from the ensemble models, using a tuned ET instead is recommended
	XGBoost	high	high	Faster and almost or even more accurate than similar methods, offers plenty of other hyperparameters, suggesting further potential in tuning
	LGBM	medium	medium	Significantly faster but less accurate than other boosting methods
	CatBoost	high	medium/high	Slow, but quite accurate, offers plenty of other hyperparameters, suggesting further potential in tuning

models (KR and MLP), the complex predictors lead to a 12% decrease of RMSE compared to the basic ones. Except for several lines of code, the complex inputs have no extra costs, as all included predictors can be calculated from the time and the three NWP variables that are already present in the basic inputs. For this reason, it is never recommended to use only the irradiance, temperature, and wind speed as inputs, but they should always be supplemented with further predictors in practical applications.

In case only the daily mean GHI is available, i.e., in the low-resolution input case, the RMSE of the best models is less than 2% higher than for the complex inputs. As daily mean GHI forecasts are typically much cheaper to obtain than full 15-min resolution NWP forecast datasets, the low-resolution case reveals a promising option for low-cost PV power forecasting, which can be highly relevant in distributed, small-scale PV systems. However, the low-resolution input case is the most sensitive to the model selection. As a general tendency, the higher the average nRMSE of a model is, the higher is the difference between the errors of the complex and low-resolution input cases. In other words, it supports that more complex models are considerably better in learning complex relationships, making the model selection especially important if only low-resolution GHI forecasts are available.

3.3. Hyperparameter tuning

The benefit of hyperparameter tuning largely depends on the model. For some models, tuning results in a high RMSE decrease of up to 47%, while for some other models, the improvement is almost zero. However, a small tuning improvement can mean either that the given model is robust, and its results depend less on the hyperparameters, or simply that the default parameters are, by fortune, well suitable for the given task. Tuning effectively reduces overfitting and extraordinary bias for models that are normally prone to these. Moreover, tuning may also improve the convergence, thus reduces the training time of some models (e.g., MLP).

The target function of the tuning is the RMSE, thus, the tuning may result in the decrease of other metrics, especially the MAE. In the case of most models, the tuning decreases RMSE and MAE simultaneously, but for some models (most notably for SVM), it results in an increased MAE. This result supports the previous finding that RMSE and MAE are two conflicting metrics due to the different statistical properties of the RMSE and MAE-optimized forecasts [75], and after a certain point, one can only be decreased at the cost of the other [80]. If the main goal is to minimize MAE, the tuning, and if possible, also the training should be performed by using MAE as the loss function. However, most high-level machine learning libraries, including *scikit-learn*, only offer the possibility of minimizing MSE during training, which makes them less suitable if the directive specifically aims to minimize MAE.

In order to facilitate the effective practical utilization of the presented models, the frequency of the best hyperparameters for the 16 PV plants in the complex tuned case is shown in Fig. 4. The actual best hyperparameters may depend on the application and data; however, the most frequent values of Fig. 4 can serve as a good starting point for further optimization in other PV modeling studies. A peak in the frequency of a hyperparameter around a specific value (i.e., the same value is the best in all PV plants, e.g., the C penalty parameter for SVM) indicates that the given value can be considered universally good for irradiance-to-power calculations. In contrast, the evenly distributed best values show that the local effects are dominant, and it is important to tune the models separately for each PV plant.

A complete overview of the effect of hyperparameter tuning for any model of interest can be obtained from the simultaneous view of Table 3 and Fig. 4. For example, as seen from Table 3, KNN has a large RMSE without tuning due to the overfitting caused by the low default number of neighbors ($k = 5$). However, this error can be largely reduced by increasing the value of k into the 100–300 range (as seen in Fig. 4), which indicates that the more averaged, smoothed power forecasts are better in terms of RMSE. Finally, it is important to note that the hyperparameter tuning in this study is limited to the at most four most relevant hyperparameters, as the number of combinations in grid search increases sharply with the increasing number and possible values of the tested hyperparameters. Still, some models (e.g., XGBoost, CatBoost, LightGBM) have even ten or more tunable hyperparameters, which indicates a space for further development.

3.4. Visual evaluation

Verification metrics can effectively summarize the model accuracy into single numbers; however, they can still not substitute the visual inspection of the forecast time series. The importance of visual inspection by a domain expert has already been emphasized by Yang [81] to ensure the best data quality, but such evaluations are rarely performed in forecasting studies.

Fig. 5 shows the power measurement and forecast time series for the three most promising models (KR, MLP, SVM) along with the GHI forecasts on randomly selected days. The measured PV power has large fluctuations in some days, which are not even captured by the GHI forecasts. Still, the ML models do not even try to add this extra variability, but on the contrary, they further smoothen the power forecasts, as the calculated power forecasts are less fluctuating than the input GHI forecasts even in the 15-min resolution complex predictor case. The power forecasts created from the low-resolution inputs are even more smooth. In this case, the shape of the daily production curves resembles the diurnal cycle of the forecasts described by the sun position angles, while the height of the daily curve is proportional to the daily mean GHI. The good accuracy of the smooth power forecasts can be attributed to two factors. First, as it has already been shown theoretically, RMSE is minimized by underdispersed forecasts [75]. Second, the fluctuation of GHI often does not coincide with the fluctuations of the PV power due to the hard predictability of the moving clouds; therefore, in partially cloudy situations issuing the expected average power as the forecast is the safest bet to avoid high errors.

Observing production curves is a hardly automatable process, but it helps to avoid small overfitting problems, which can be easily corrected manually. As an example, it can be seen in Figs. 5 and 6 that at some weather conditions, specifically at low irradiance (e.g. 2019-11-20, 2019-04-12), even KR badly overestimates the convex parts (around sunrise and sunset), specifically in the low-resolution input case. A similar tendency is also visible for SVM, which can be a possible explanation for its high positive bias (see Table 3), as the lack of near-zero values contributes to increasing the mean of the forecasts. This kind of error can be fixed either manually by e.g., rescaling the forecasts in the problematic intervals or automatically by clipping the results by time-dependent clear-sky maximum power outputs instead of a constant absolute maximum. However, the highest possible PV power at a given time can only be derived by applying a physical PV model on clear-sky irradiance forecasts [82], which is impossible in many cases when the cause for using ML is the insufficient knowledge of the PV design data.

4. Conclusions and recommendations

This paper presents a comparison of 24 easy-to-applicable machine learning (ML) models for day-ahead photovoltaic (PV) power forecasting based on numerical weather predictions (NWP). The models are tested for two-years data of 16 Hungarian ground-mounted PV plants and evaluated by five commonly used verification metrics. In addition to the models, the selection of the predictors and the significance of the hyperparameter are also examined. The main findings of this study are summarized as follows:

- The overall most accurate model is *kernel ridge* (KR), but it has an extremely long training time and high memory usage. The *multilayer perceptron* (MLP, the basic type of feedforward neural networks) has an almost identical accuracy with significantly lower training time; thus, this model is recommended for practical applications. However, KR is worthy for further, more specified investigations.
- Input data selection is even more important than model selection. Extending the basic NWP outputs with the angles calculated from the position of the Sun (elevation, azimuth, and declination) and the time-shifted and averaged versions of the global horizontal irradiance (GHI) leads to an RMSE reduction of 13.1% compared to the basic input case that only uses the GHI, ambient temperature and wind speed as predictors. Moreover, using only daily mean GHI forecasts and the Sun position angles as predictors increases RMSE by only 1.5% compared to the best input case that relies on a full set of 15-min GHI forecasts. In other words, the added value of high-resolution irradiance forecasts is limited for PV power forecasting in a day-ahead time horizon.
- Hyperparameter tuning is essential to exploit the full potential of the ML models, as the best RMSE achieved without tuning is 3.1% higher than the best with tuning. The more robust models (e.g., MLP, *gradient boosting regressor*) are quite accurate even without tuning, while for some other models (e.g., *lasso*, *decision tree*), training is a must, as they have enormous errors with their default parameters. Overall, tuning equalizes the RMSE of the different models compared to the default case.
- The overall best results (tuned KR or MLP with the complex predictors) has 13.9% lower RMSE than the simplest solution, i.e., the *linear regression* fitted on the basic NWP data. The importance of the ML model and input selection falls into the same range as model chain selection for physical irradiance-to-power conversion [10].

The qualitative description of the accuracy and hyperparameter sensitivity of the 24 tested models, supplemented by further notes and recommendations, are summarized in Table 4. The best hyperparameter values shown in Fig. 4 can serve as starting points in practical applications. Even though this study deals with NWP-based irradiance-to-power conversion, the results are expected to be similar for other types of irradiance forecasts, as the task of the ML models, i.e., describing the operation of the PV plants, is the same regardless of the source of the irradiance data.

This paper focuses on the comparison of application-ready ML models from a high-level program library; thus, it serves as a guide for researchers and practitioners for providing reasonably good solutions for operational PV forecasting. However, it still offers room for further development of more accurate irradiance-to-power conversion methods. One possible direction is the use of more advanced ML models, including the nowadays popular deep learning methods that are also based on neural networks similarly to MLP, the overall best-recommended model of this paper [42,83]. Another possibility is to dig deeper into the hybridization of physical and machine learning models, as our results revealed that adding even such simple theoretically calculated data to the predictors as the Sun position angles have significantly enhanced the accuracy of the power predictions. Finally, extending the study to probabilistic photovoltaic power forecasting and testing the models in such context is also an interesting topic for the future.

Credit author statement

Dávid Markovics: Conceptualization, Methodology, Software, Validation, Formal analysis, Investigation, Data curation, Writing – original draft. Martin János Mayer: Conceptualization, Methodology, Validation, Data curation, Writing – original draft, Writing – review & editing, Supervision.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgment

The authors express their gratitude to Gabriella Szépszó and Mihály Szűcs from the Hungarian Meteorological Service for providing the AROME numerical weather prediction data, and Norbert Péter and Róbert Csapó from the MVM Green Generation Ltd. for the PV

production data. The research reported in this paper and carried out at BME has been supported the ÚNKP-21-2 New National Excellence Program and the TKP2020 National Challenges Subprogram (Grant No. BME-NCS) of the Ministry for Innovation and Technology from the source of the National Research, Development, and Innovation Fund (Grant Nos. ÚNKP-21-2, BME-NCS).

References

[1] Fleiter T, Herbst A, Rehfeldt M, Arens M. Industrial Innovation: pathways to deep decarbonisation of Industry. Part 2: scenario analysis and pathways to deep decarbonisation. n.d.
[2] International Energy Agency. Renewables 2020 – analysis - IEA. Iea; 2020.
[3] Hong T, Pinson P, Wang Y, Weron R, Yang D, Zareipour H. Energy forecasting: a review and outlook. IEEE Open Access Journal of Power and Energy 2020;7: 376–88. https://doi.org/10.1109/OAJPE.2020.3029979.
[4] Yang D, Kleissl J, Gueymard CA, Pedro HTC, Coimbra CFM. History and trends in solar irradiance and PV power forecasting: a preliminary assessment and review using text mining. Sol Energy 2018;168:60–101. https://doi.org/10.1016/j.solener.2017.11.023.
[5] Hong T, Pinson P, Fan S, Zareipour H, Troccoli A, Hyndman RJ. Probabilistic energy forecasting: global energy forecasting competition 2014 and beyond. Int J Forecast 2016;32:896–913. https://doi.org/10.1016/j.ijforecast.2016.02.001.
[6] Yang D. A guideline to solar forecasting research practice: reproducible, operational, probabilistic or physically-based, ensemble, and skill (ROPES). J Renew Sustain Energy 2019;11:20. https://doi.org/10.1063/1.5087462.
[7] Yang D, Alessandrini S, Antonanzas J, Antonanzas-Torres F, Badescu V, Beyer HG, et al. Verification of deterministic solar forecasts. Sol Energy 2020;210:20–37. https://doi.org/10.1016/j.solener.2020.04.019.
[8] Yang D, van der Meer D. Post-processing in solar forecasting: ten overarching thinking tools. Renew Sustain Energy Rev 2021;140:110735. https://doi.org/10.1016/j.rser.2021.110735.
[9] Yagli GM, Yang D, Srinivasan D. Automatic hourly solar forecasting using machine learning models. Renew Sustain Energy Rev 2019;105:487–98. https://doi.org/10.1016/j.rser.2019.02.006.
[10] Mayer MJ, Gróf G. Extensive comparison of physical models for photovoltaic power forecasting. Appl Energy 2021;283:116239. https://doi.org/10.1016/j.apenergy.2020.116239.
[11] Blaga R, Sabadus A, Stefu N, Dughir C, Paulescu M, Badescu V. A current perspective on the accuracy of incoming solar energy forecasting. Prog Energy Combust Sci 2019;70:119–44. https://doi.org/10.1016/j.pecs.2018.10.003.
[12] Murphy AH, Epstein ES. Skill scores and correlation coefficients in model verification. Mon Weather Rev 1989;117:572–82. https://doi.org/10.1175/1520-0493(1989)117<0572:SSACCI>2.0.CO;2.
[13] Marquez R, Coimbra CFM. Proposed metric for evaluation of solar forecasting models. J Sol Energy Eng 2013;135:1–9. https://doi.org/10.1115/1.4007496.
[14] Yang D. Standard of reference in operational day-ahead deterministic solar forecasting. J Renew Sustain Energy 2019;11:053702. https://doi.org/10.1063/1.5114985.
[15] Yang D, Wu E, Kleissl J. Operational solar forecasting for the real-time market. Int J Forecast 2019;35:1499–519. https://doi.org/10.1016/j.ijforecast.2019.03.009.
[16] Ulbricht R, Fischer U, Lehner W, Donker H. First steps towards a systematical optimized strategy for solar energy supply forecasting. In: European conference on machine learning and principles and practice of knowledge discovery in databases, ECMLPKDD 2013; 2013.
[17] Lorenz E, Scheidsteger T, Hurka J. Regional PV power prediction for improved grid integration. Prog Photovoltaics Res Appl 2011;19:757–71. https://doi.org/10.1002/pip.1033.
[18] Mayer MJ. Influence of design data availability on the accuracy of physical photovoltaic power forecasts. Sol Energy 2021;227:532–40. https://doi.org/10.1016/j.solener.2021.09.044.
[19] Antonanzas J, Osorio N, Escobar R, Urraca R, Martinez-de-Pison FJ, Antonanzas-Torres F. Review of photovoltaic power forecasting. Sol Energy 2016;136:78–111. https://doi.org/10.1016/j.solener.2016.06.069.
[20] Yang D, Dong Z. Operational photovoltaics power forecasting using seasonal time series ensemble. Sol Energy 2018;166:529–41. https://doi.org/10.1016/j.solener.2018.02.011.
[21] Yang X, Xu M, Xu S, Han X. Day-ahead forecasting of photovoltaic output power with similar cloud space fusion based on incomplete historical data mining. Appl Energy 2017;206:683–96. https://doi.org/10.1016/j.apenergy.2017.08.222.
[22] de Giorgi MG, Congedo PM, Malvoni M. Photovoltaic power forecasting using statistical methods: impact of weather data. IET Sci Meas Technol 2014;8:90–7. https://doi.org/10.1049/iet-smt.2013.0135.
[23] Malvoni M, de Giorgi MG, Congedo PM. Forecasting of PV Power Generation using weather input data-preprocessing techniques. Energy Proc 2017;126:651–8. https://doi.org/10.1016/j.egypro.2017.08.293. Elsevier Ltd.
[24] Sheng H, Xiao J, Cheng Y, Ni Q, Wang S. Short-term solar power forecasting based on weighted Gaussian process regression. IEEE Trans Ind Electron 2018;65:300–8. https://doi.org/10.1109/TIE.2017.2714127.
[25] 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; 183:267–82. https://doi.org/10.1016/J.RENENE.2021.10.102.
[26] Qu Y, Xu J, Sun Y, Liu D. A temporal distributed hybrid deep learning model for day-ahead distributed PV power forecasting. Appl Energy 2021;304:117704. https://doi.org/10.1016/J.APENERGY.2021.117704.
[27] AlShafeey M, Csáki C. Evaluating neural network and linear regression photovoltaic power forecasting models based on different input methods. Energy Rep 2021;7:7601–14. https://doi.org/10.1016/J.EGYR.2021.10.125.
[28] Zhao W, Zhang H, Zheng J, Dai Y, Huang L, Shang W, et al. A point prediction method based automatic machine learning for day-ahead power output of multi-region photovoltaic plants. Energy 2021;223. https://doi.org/10.1016/j.energy.2021.120026.
[29] Wang F, Xuan Z, Zhen Z, Li K, Wang T, Shi M. A day-ahead PV power forecasting method based on LSTM-RNN model and time correlation modification under partial daily pattern prediction framework. Energy Convers Manag 2020;212: 112766. https://doi.org/10.1016/J.ENCONMAN.2020.112766.
[30] Miraftabzadeh SM, Longo M, Foiadelli F. A-day-ahead photovoltaic power prediction based on long short term memory algorithm. In: Sest 2020 - 3rd international conference on smart energy systems and technologies; 2020. https://doi.org/10.1109/SEST48500.2020.9203481.
[31] Liu L, Zhan M, Bai Y. A recursive ensemble model for forecasting the power output of photovoltaic systems. Sol Energy 2019;189:291–8. https://doi.org/10.1016/j.solener.2019.07.061.
[32] Lin P, Peng Z, Lai Y, Cheng S, Chen Z, Wu L. Short-term power prediction for photovoltaic power plants using a hybrid improved Kmeans-GRA-Elman model based on multivariate meteorological factors and historical power datasets. Energy Convers Manag 2018;177:704–17. https://doi.org/10.1016/j.enconman.2018.10.015.
[33] Raza MQ, Mithulananthan N, Summerfield A. Solar output power forecast using an ensemble framework with neural predictors and Bayesian adaptive combination. Sol Energy 2018;166:226–41. https://doi.org/10.1016/j.solener.2018.03.066.
[34] Pierro M, de Felice M, Maggioni E, Moser D, Perotto A, Spada F, et al. Photovoltaic generation forecast for power transmission scheduling: a real case study. Sol Energy 2018;174:976–90. https://doi.org/10.1016/j.solener.2018.09.054.
[35] Wang J, Ran R, Song Z, Sun J. Short-term photovoltaic power generation forecasting based on environmental factors and GA-SVM. Journal of Electrical Engineering and Technology 2017;12:64–71. https://doi.org/10.5370/JEET.2017.12.1.064.
[36] Tascikaraoglu A, Sanandaji BM, Chicco G, Cocina V, Spertino F, Erdinc O, et al. Compressive spatio-temporal forecasting of meteorological quantities and photovoltaic power. IEEE Trans Sustain Energy 2016;7:1295–305. https://doi.org/10.1109/TSTE.2016.2544929.
[37] Wolff B, Kühnert J, Lorenz E, Kramer O, Heinemann D. Comparing support vector regression for PV power forecasting to a physical modeling approach using measurement, numerical weather prediction, and cloud motion data. Sol Energy 2016;135:197–208. https://doi.org/10.1016/j.solener.2016.05.051.
[38] Mellit A, Kalogirou SA. Artificial intelligence techniques for photovoltaic applications : a review. Prog Energy Combust Sci 2008;34:574–632. https://doi.org/10.1016/j.pecs.2008.01.001.
[39] Yang D, Li W, Yagli GM, Srinivasan D. Operational solar forecasting for grid integration: standards, challenges, and outlook. Sol Energy 2021;224:930–7. https://doi.org/10.1016/j.solener.2021.04.002.
[40] Szintai B, Szűcs M, Randriamampianina R, Kullmann L. Application of the AROME non-hydrostatic model at the Hungarian meteorological service: physical parameterizations and ensemble forecasting. Idojaras 2015;119:241–65.
[41] Reda I, Andreas A. Solar position algorithm for solar radiation applications. Sol Energy 2004;76:577–89. https://doi.org/10.1016/j.solener.2003.12.003.
[42] Ahmed R, Sreeram V, Mishra Y, Arif MD. A review and evaluation of the state-of-the-art in PV solar power forecasting: techniques and optimization. Renew Sustain Energy Rev 2020;124:109792. https://doi.org/10.1016/J.RSER.2020.109792.
[43] Voyant C, Notton G, Kalogirou S, Nivet ML, Paoli C, Motte F, et al. Machine learning methods for solar radiation forecasting: a review. Renew Energy 2017; 105:569–82. https://doi.org/10.1016/j.renene.2016.12.095.
[44] Alpaydin E. Introduction to machine learning. second ed. The MIT Press; 2010.
[45] Pedregosa F, Varoquaux G, Gramfort A, Michel V, Thirion B, Grisel O, et al. Scikit-learn: machine learning in Python. J Mach Learn Res 2011;12:2825–30.
[46] Friedman J, Hastie T, Tibshirani R. Regularization paths for generalized linear models via coordinate descent. J Stat Software 2010;33:1–22.
[47] Rifkin RM, Lippert RA. Notes on regularized least squares. Computer Science and Artificial Intelligence Laboratory Technical Report - MIT-CSAIL-TR-2007-025 2007.
[48] Efron B, Hastie T, Johnstone I, Tibshirani R. Least angle regression. Ann Stat 2002; 32:407–99. https://doi.org/10.1007/978-0-387-75692-9_20.
[49] Rubinstein R, Zibulevsky M, Elad M. Efficient implementation of the K-SVD algorithm using batch orthogonal matching pursuit. CS Technion 2008;1–15.
[50] MacKay DJC. Bayesian interpolation. Neural Comput 1992;4:415–47. https://doi.org/10.1162/neco.1992.4.3.415.
[51] Wipf D, Nagarajan S. A New view of automatic relevance determination. Adv Neural Inf Process Syst 2007;20.
[52] Crammer K, Dekel O, Keshet J. Online passive-aggressive algorithms shai shalev-shwartz yoram singer †, vol. 7; 2006.
[53] Choi S, Kim T, Yu W. Performance evaluation of RANSAC family. 2009. https://doi.org/10.5244/C.23.81.
[54] Dang X, Peng H, Wang X, Zhang H. Theil-sen estimators in a multiple linear regression model. 2009.
[55] Owen AB. A robust hybrid of lasso and ridge regression. Contemp Math 2007;443. https://doi.org/10.1090/conm/443.

[56] Vovk V. kernel ridge regression. In: Schölkopf B, Luo Z, Vovk V, editors. Empirical inference: festschrift in honor of vladimir N. Vapnik. Berlin, Heidelberg: Springer Berlin Heidelberg; 2013. p. 105–16. https://doi.org/10.1007/978-3-642-41136-6_11.
[57] Chang C-C, Lin C-J. LIBSVM: a library for support vector machines. 2001.
[58] Kingma DP, Lei Ba J. Adam: a method for stochastic optimization. 2015.
[59] Kramer O K-Nearest Neighbors. Dimensionality reduction with unsupervised nearest neighbors. Berlin, Heidelberg: Springer Berlin Heidelberg; 2013. p. 13–23. https://doi.org/10.1007/978-3-642-38652-7_2.
[60] Xu M, Watanachaturaporn P, Varshney PK, Arora MK. Decision tree regression for soft classification of remote sensing data. Remote Sens Environ 2005;97:322–36. https://doi.org/10.1016/j.rse.2005.05.008.
[61] Breiman L. Random forests. Mach Learn 2001;45:5–32. https://doi.org/10.1023/A:1010933404324. 2001 45:1.
[62] Geurts P, Ernst D, Wehenkel L. Extremely randomized trees. Mach Learn 2006;63:3–42. https://doi.org/10.1007/s10994-006-6226-1.
[63] Freund Y, Schapire RE. A decision-theoretic generalization of on-line learning and an application to boosting. J Comput Syst Sci 1997;55:119–39. https://doi.org/10.1006/jcss.1997.1504.
[64] Friedman JH. Greedy function approximation: a gradient boosting machine. Ann Stat 2001;29:1189–232. https://doi.org/10.1214/aos/1013203451.
[65] XgboostDevelopers. Xgboost docs 2020. https://xgboost.readthedocs.io/en/latest/index.html.
[66] Microsoft C. LightGBM docs 2021. https://lightgbm.readthedocs.io/en/latest/Features.html.
[67] Prokhorenkova L, Gusev G, Vorobev A, Dorogush AV, Gulin A. Catboost: unbiased boosting with categorical features. Adv Neural Inf Process Syst 2018;2018-Decem:6638–48.
[68] Agrawal T. Introduction to hyperparameters. Hyperparameter Optimization in Machine Learning 2021:1–30. https://doi.org/10.1007/978-1-4842-6579-6_1.
[69] Agrawal T. Hyperparameter optimization using scikit-learn. Hyperparameter Optimization in Machine Learning 2021:31–51. https://doi.org/10.1007/978-1-4842-6579-6_2.
[70] Snoek J, Larochelle H, Adams RP. Practical bayesian optimization of machine learning algorithms. 2012.
[71] Jamieson K, Talwalkar A. Non-stochastic best arm identification and hyperparameter optimization. 19th international conference on artificial intelligence and statistics. AISTATS); 2015.
[72] Hertel L, Collado J, Sadowski P, Ott J, Baldi P. Sherpa: robust hyperparameter optimization for machine learning. Software 2020;12:100591. https://doi.org/10.1016/J.SOFTX.2020.100591.
[73] Khalid R, Javaid N. A survey on hyperparameters optimization algorithms of forecasting models in smart grid. Sustain Cities Soc 2020;61:102275. https://doi.org/10.1016/J.SCS.2020.102275.
[74] Kolassa S. Why the "best" point forecast depends on the error or accuracy measure. Int J Forecast 2020;36:208–11. https://doi.org/10.1016/j.ijforecast.2019.02.017.
[75] Mayer MJ, Yang D. Calibration of deterministic NWP forecasts and its impact on verification. Int J Forecast 2020 [submitted for publication].
[76] Antonanzas J, Perpinan-Lamigueiro O, Urraca R, Antonanzas-Torres F. Influence of electricity market structures on deterministic solar forecasting verification. Sol Energy 2020;1–3. https://doi.org/10.1016/j.solener.2020.04.017.
[77] ENTSO-E. (European Network of, Transmission System Operators, for Electricity). An overview of the European balancing market and electricity balancing guideline. 2018.
[78] Gneiting T. Making and evaluating point forecasts. J Am Stat Assoc 2011;106:746–62. https://doi.org/10.1198/jasa.2011.r10138.
[79] Gijsbers P, LeDell E, Thomas J, Poirier S, Bischl B, Vanschoren J. An open source AutoML benchmark. 6th ICML Workshop on Automated Machine Learning; 2019. p. 1–8.
[80] Mayer MJ. Design optimization and power forecasting of photovoltaic power plants. Budapest University of Technology and Economics; 2020. https://doi.org/http://hdl.handle.net/10890/15112.
[81] Yang D. Validation of the 5-min irradiance from the national solar radiation database (NSRDB). J Renew Sustain Energy 2021;13:016101. https://doi.org/10.1063/5.0030992.
[82] Engerer NA, Mills FP. KPV: a clear-sky index for photovoltaics. Sol Energy 2014;105:679–93. https://doi.org/10.1016/j.solener.2014.04.019.
[83] Wang K, Qi X, Liu H. A comparison of day-ahead photovoltaic power forecasting models based on deep learning neural network. Appl Energy 2019;251. https://doi.org/10.1016/j.apenergy.2019.113315.