Individual yield nowcasting for residential PV systems

Grzebyk, Daniel ; Alcañiz , Alba; Donker, Jaap; Zeman, Miro; Ziar, Hesan; Isabella, Olindo

DOI
10.1016/j.solener.2023.01.036

Publication date
2023

Document Version
Final published version

Published in
Solar Energy

Solar Energy 251 (2023) 325–336

Contents lists available at ScienceDirect

Solar Energy

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

Individual yield nowcasting for residential PV systems

Daniel Grzebyk [a,b], Alba Alcañiz [a,*], Jaap C.B. Donker [b], Miro Zeman [a], Hesan Ziar [a], Olindo Isabella [a]

[a] Photovoltaic Materials and Devices Group, Delft University of Technology, Mekelweg 4, Delft, 2628 CD, The Netherlands
[b] Solar Monkey B.V., Wilhelmina van Pruisenweg 35, Den Haag, 2595 AN, The Netherlands

ARTICLE INFO

Keywords:
Photovoltaics
Solar yield prediction
Forecasting
Nowcasting
Machine learning
Loss function
XGBoost

ABSTRACT

Due to the inherent uncertainty in photovoltaic (PV) energy generation, an accurate power forecasting is essential to ensure a reliable operation of PV systems and a safe electric grid. Machine learning (ML) techniques have gained popularity on the development of this task due to its increased accuracy. Most literature, however, focuses only on less than 5 PV systems during training process, which does not ensure generalization to unseen systems. When in presence of a large feet, regional forecasts are the norm. Nevertheless, none of these approaches are usable when it comes to monitoring residential PV systems. In this work, we propose a single ML model that is able to predict the individual power of a large fleet of 1102 PV systems. XGBoost algorithm was selected as the most suitable algorithm for the task of PV yield nowcasting due to its performance and ease of use. This algorithm obtains Mean Absolute Error (MAE) of 0.877 kWh (considering an average system size of 4.44 kWp) and Mean Absolute Percentage Error (MAPE) of 23% for hourly data aggregated to daily values. XGBoost predictions for individual PV systems are on average two times better than currently used commercial software. We discuss the lack of a suitable loss function that can combine absolute and relative errors for residential PV yield forecasting. We also point out the lack of an adequate metric to compute the error made on the predictions and provide hints on developing a suitable one.

1. Introduction

Between 2010 and 2020, cumulative photovoltaic (PV) installations grew at a compound rate of 34% annually (Philipps et al., 2021) being one of the most promising technologies to tackle climate change. With economic growth being the crucial factor shaping human actions and policies, solar energy can have a profound impact worldwide. Solar power is abundant, affordable, easily scalable and has small CO_2 emissions associated with manufacturing (Photovoltaic Power Systems Program, 2021). However, mass utilization of solar modules has a major challenge of intermittency of supply, which complicates power balance maintenance, reserve capacity planning and electricity market bidding (Notton et al., 2018).

A way to help system operators to manage the grid is to inform them how much electricity will be generated by these renewable sources. Consequently, PV yield forecasting is an important factor facilitating energy transition and supporting investment in solar energy. Accurate forecasts decrease energy yield uncertainty, therefore reducing generation-load mismatch in the power grid and decreasing investment risk. Yield nowcasting (short-term forecasting) ensures early anomaly detection preventing financial losses and contributing to security of PV system owners.

The PV power forecasting techniques can be classified into three major groups (Ahmed et al., 2020): persistence, physical and statistical models. The latter group can be further divided into time-series based and machine learning (ML) techniques. Until the surge of ML techniques, none of the approaches rose to the forecasting challenge in terms of reliability, accuracy and computational economy (Ahmed et al., 2020). Persistence techniques are too simple to provide accurate results except for in the very short-term. Analytical equations often fail to predict yield due to incomplete system information or insufficient spatial and temporal resolution of weather predictions (Antonanzas et al., 2016). Moreover, in case of residential PV systems taking continuous measurements of all required parameters in situ is not a common practice due to high associated costs. Finally the main drawback of statistical methods is their inability to adapt to abrupt changes in the meteorological conditions (Hossain et al., 2017).

Overall, the obvious choice is the machine learning group. Although some algorithms present drawbacks such as reduced reliability due to randomness, increased model complexity, difficulty of hyperparameter tuning and large data set requirement (Akhter et al., 2019; Das et al., 2018), the increase in accuracy at a reduced computational

* Corresponding author.
E-mail address: A.AlcanizMoya@tudelft.nl (A. Alcañiz).

https://doi.org/10.1016/j.solener.2023.01.036
Received 24 August 2022; Accepted 20 January 2023
Available online 24 January 2023

time balances them out. Some interesting algorithms applied for PV power forecasting are Multivariate Adaptive Regression Splines (Li et al., 2016), Random forest (Ferlito et al., 2017; Massaoudi et al., 2019), Long–short term memory (Han et al., 2019), Extreme Learning Machine (Ni et al., 2017) or hybrid ML techniques (Wang et al., 2019). ML algorithms have shown to provide accurate forecasts in a wide range of horizons, from the very short (Sun et al., 2019a; Anagnostos et al., 2019) to the long (Lin and Pai, 2016a; Jung et al., 2020) term.

Despite overwhelming abundance of literature, there are still gaps in this discipline. For instance, Yang found a lack of standard benchmark, lack of open-source access to the utilized data, intentional hiding of the shortcomings and evaluation on small data sets (Yang, 2019). The latter was also noticed by Theocharides et al. (2018). Most researchers motivate their study by contributing to improved generator dispatch, power quality effects mitigation, and reducing secondary reserve capacity, but if their results are not validated for multiple systems, they might not be reliable enough to upscale. Few studies employed data from more than 5 PV systems, and when they did, PV power was commonly aggregated to provide a regional approach (Lin and Pai, 2016b; Agoua et al., 2019; De Felice et al., 2015; Fu et al., 2019; Lorenz et al., 2011; Wolff et al., 2016; Pierro et al., 2017). Although from a grid manager perspective regional forecasts are preferred over individual ones due to their lower errors (Zhang et al., 2022), when it comes to nowcasting and monitoring the individual PV systems, a regional forecast is not suitable.

Exceptions found in literature are (Elsinga and van Sark, 2017), which provided analysis for 202 rooftop PV systems, Gensler et al. who forecasted the power of 21 systems (Gensler et al., 2017), and the forecast of 71 systems from Lee et al. (2018). The approach in these cases consisted on creating an ML algorithm for each PV system, which can be computationally intensive during the training procedure and can limit the application of the ML model until enough data is available for a new PV system.

As opposed to previous literature, the aim of this work is to employ a single ML algorithm to make accurate and fast predictions for all PV systems of a large database of 1102 PV systems at once. The algorithm will benefit from the training of other systems, its training time will be reduced, and it will distinguish between systems through easily available system information. The developed methodology will be applied in a solar software company to show an alternative to the currently employed analytical-based model. Suggestions for developing a standard metric are also given, as well as a discussion on loss functions. This report is structured as follows. Section 2 details the dataset employed in this work, focusing on the considered features, data preparation and exploration. Section 3 describes the methods employed. The employed feature selection strategy is described, as well as the ML models. Two important characteristics of the methodology, namely the loss functions and the metrics, are also discussed in order to highlight their flaws and possible improvements. Results are explained in Section 4 in terms of feature selection, models performance, learning curves and error analysis. Finally, main conclusions are presented in Section 5.

2. Dataset

In this section origin, preparation and main characteristics of input data are explained. The first step is to gather all possible data for developing the model. Before inputting it to a ML algorithm, data needs to be cleaned and processed. The data set is also explored to determine main data quality issues.

2.1. Considered features

All considered features for the machine learning model can be divided into weather and system features and descriptive parameters. In machine learning, *features* is the term for the variables input to a model. All features are available in the time period from July 1st 2018 until June 30th 2019 for 1102 PV systems located in the Netherlands and Belgium. Overall, this study considers a large data set containing more than 4,000,000 data points.

Several weather parameters were considered for the model. These include: global horizontal irradiance (GHI), cloud coverage, wind speed, precipitation and ambient temperature. This data was obtained from the Royal Dutch Meteorological Institute with hourly resolution (Anon, 2021a). The sun angles (altitude and azimuth) were included as well using the library pvlib (Holmgren et al., 2018).

The system parameters refer to the characteristics of the PV systems and include: system size, age, type of panels, latitude, longitude, panel inclination, orientation, decay per year, maximal inverter efficiency and nominal operating cell temperature. Descriptive parameters consist of day of year and historical yield from 24, 48, 72, 96 and 120 h before. Decision to use only five past days was based on Maitanova et al. (2020), but a longer horizon could have been utilized as well. All system data has been provided by the start-up Solar Monkey with headquarters in The Hague, the Netherlands (Anon, 2021b).

2.2. Data preparation

Data needs to be properly prepared for the machine learning model in order to obtain the best results. The data set was explored in search for quality issues. For instance, missing cloud coverage values were replaced with mean cloud coverage across all samples equal to 5.83 okta. Samples corresponding to night values (zero GHI) were removed from the data set. Several issues were found on the yield data obtained from inverters. Inverters log hourly yields via wi-fi which might be discontinuous. In case of prolonged disconnection, the amount of data stored in an inverter might exceed its memory capacity and some information is lost leading to missing yield values. Other issues are constant or lagged yields. Constant yields might be caused by unreported disconnection of the inverter. Examples of these data quality issues can be seen in Fig. 1 where low outliers are below 0.13 percentile and high outliers are above 99.87 percentile of all yield values.

Another step of data preprocessing is normalization used to obtain similar magnitudes of all features. This is required by ML algorithms to provide optimal performance. First, presented inputs were processed to contain only numerical features. Then, each of the features were scaled to values between 0 and 1 using the function *MinMaxScaler* from *sklearn* Python package (Pedregosa et al., 2011). This scaler uses minimum and maximum values of all hourly data calculated for every feature separately.

2.3. Data exploration

Brief exploration of the data was conducted as well. It was discovered that nearly 79% of systems consist of mono-crystalline silicon, 18% of poly-crystalline silicon and around 3% of thin film panels. Systems of size between 2.5 kWp and 7.5 kWp dominate, but larger systems, up to 17.7 kWp, occur in the data set. None of the analyzed systems was older than four years.

Regarding weather features, GHI is dominated by small values below 50 W/m^2 and its maximal values do not exceed 950 W/m^2. Visualization of GHI vs. day of year and hour can be seen in Fig. 2. According to cloud coverage data, sky is almost completely overcast for vast majority of time (around 70%). Ambient temperatures remain between 264.2 K (−9 °C) and 310.2 K (37.05 °C) which is reasonable. Rainfall is present in 13.4% of all samples and its maximal value equals 21.6 mm per hour or 37.14 mm per day.

Exploration continued by looking for correlation in the data. Features used as inputs to ML models should not be correlated and should have possibly large variance (Bishop, 2006). Correlation was investigated by plotting a heatmap of weather features visible in Fig. 3.

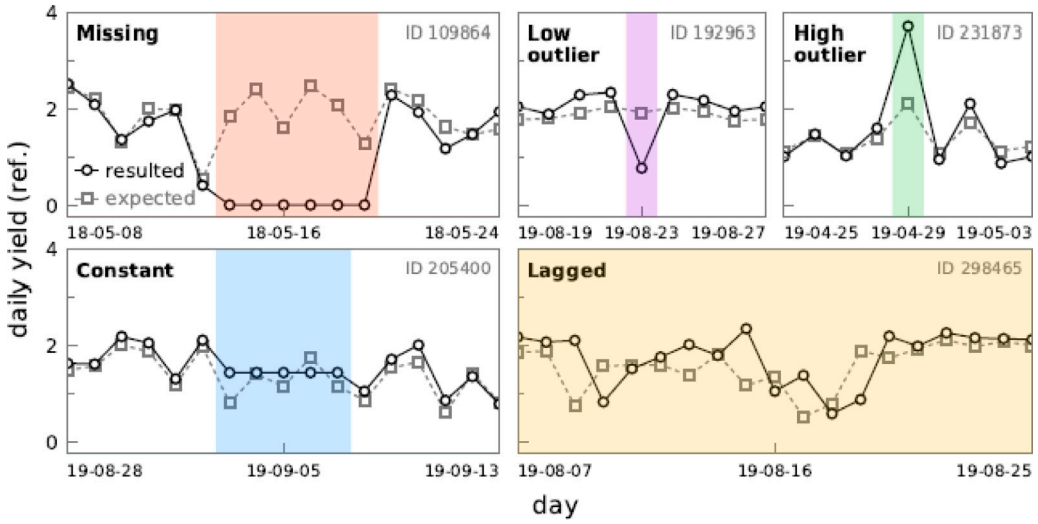

Fig. 1. Examples of yield data quality issues.

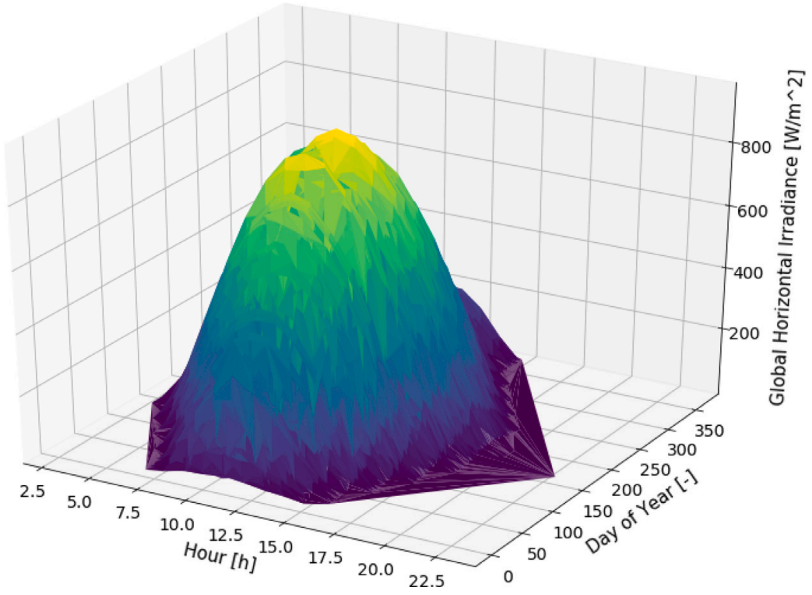

Fig. 2. Distribution of global horizontal irradiance values along the day and year.

It can be seen that GHI is positively correlated with sun altitude and negatively correlated with cloud coverage. That is intuitive, as clouds cause shading and reduce irradiance incident on flat surface on the ground. Sun altitude has strong positive correlation with ambient temperature. That is understandable, as the higher the sun is in the sky, the more irradiance reaches Earth surface and the hotter the air becomes. Ambient temperature is correlated with day of year since it incorporates weather seasonality.

3. Methodology

This section explains the methods employed for selecting the best subset of features and predicting the PV power. It also offers a discussion on loss functions and metrics, important parameters for PV power prediction.

3.1. Feature selection

Given the high amount of features considered, it is important to determine which are the relevant inputs for the ML models. Feature selection is a process of eliminating the inputs which do not contribute to increasing algorithm's performance (Konstantinos Koutrombas, 2008). This will decrease the computational time while facilitating the models' implementation. Amongst the default *sklearn* feature selection functions available, Recursive Feature Elimination (RFE) with 3-fold cross validation was selected. This technique was used to increase confidence that the selected set of features is the best possible, so that maximum performance could be achieved.

As its name indicates, RFE eliminates features recursively given a certain metric and algorithm (Guyon et al., 2002). The function computes feature importance, drops the feature with the lowest importance and recalculates the metric. The process is iterated until only a user-specific number of features is left. The importance of each feature is calculated by the chosen algorithm (Granitto et al., 2006). Each training process is repeated three times using different parts of the data to make sure the results are independent of shuffling. Once the best configuration is found, it is implemented in the ML models and used for prediction of PV power output.

The selected algorithm strongly influences the RFE results. In this case, the algorithm used was XGBoost and the feature was Root Mean Squared Error (RMSE). Since XGBoost is made of a combination of

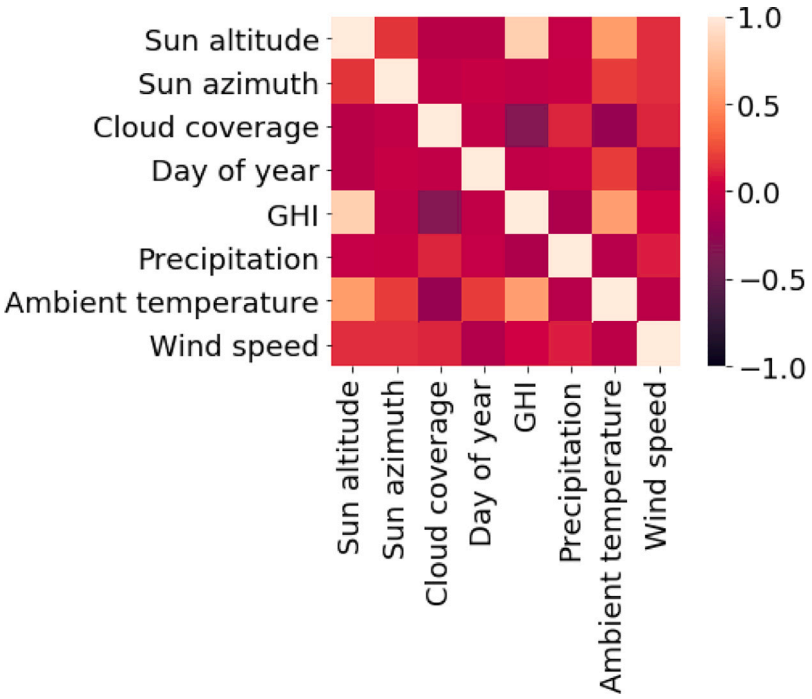

Fig. 3. Correlation heatmap of the weather variables using hourly resolution.

decision trees (more information in the next section), the importance of each feature is relatively straightforward to compute. A decision tree makes predictions by splitting decisions into branches. The data space is recursively partitioned depending on its value, and a simple prediction model is then fitted within each partition (Loh, 2011). Importance is a score that indicates how useful each feature is in the construction of a tree. It is calculated based on the amount that each feature improves the performance in each split, weighted by the number of observations on the split, and averaged across all trees (Brownlee, 2016).

3.2. Models

This section briefly explains the working principle of Extreme Gradient Boosting (XGBoost), the main model employed in this publication. It also describes persistence and Solar Monkey's model. Persistence is a baseline model commonly employed to compute the skill score (vide Section 3.4). Solar Monkey's model is the commercial model employed by the start-up, and used here as a reference physics-based model.

Persistence model assumes that the PV power at the following time step will be the same as the previous one (Kleissl, 2013). It is a simple algorithm usually employed as benchmark to compare the performance of other models. Eq. (1) describes the model, where $P(t)$ is the PV power produced at time step t. Several improvements to this simple algorithm have been developed to improve the accuracy, such as persistence based on solar power index (Antonanzas et al., 2016). However, in this study the persistence model is in its most naive form. Despite its simplicity, this model can outperform complex algorithms when forecasting PV power in the short-term (Sun et al., 2019b).

$$P(t) = P(t-1) \tag{1}$$

The algorithm currently employed by Solar Monkey to compute the PV power produced by a PV system is based on an analytical approach. It makes use of the skyline profile to account for obstacles surrounding PV modules and the sun movement. Details on the framework can be found in de Vries et al. (2020).

Regarding the choice of ML algorithm, it was decided to opt for an algorithm whose performance was already proved in previous works. This algorithm should be able to handle large amounts of data in a fast and accurate way, and it should be of reduced complexity and easy to optimize. Overall, it was decided to select an algorithm from the ensembles' family due to their ability to win ML competitions (Vorhies, 2016; Chen and Guestrin, 2016). For instance, Gradient Boosting was employed in GEFCom competition when predicting solar power by three out of five top participants (Hong et al., 2016). Inside the family of ensembles, the algorithm of choice was extreme gradient boosting (XGBoost). This algorithm has already been employed for solar yield forecasting in the works (Ferlito et al., 2017; Isaksson and Conde, 2018; Massaoudi et al., 2019). Ferlito et al. (2017) employed XGBoost together with 10 other prediction models to forecast the yield of a 1 kW grid-connected PV system. Their results indicated that support vector regression was the best model out of 11 investigated, unless the dataset was limited to less than 60 days, in which case ensemble algorithms were the best alternatives. In Isaksson and Conde (2018), a comparison of the performance of several ML models for PV power prediction was performed across five different sites in Sweden. Their main conclusion was that Artificial Neural Networks and Gradient Boosting Regression Trees performed best on average across all sites. Finally, Massaoudi et al. (2019) improved XGBoost by combining it with other algorithms.

Extreme Gradient Boosting, or XGBoost is an ensemble machine learning algorithm based on decision trees. Decision trees tend to overfit, since they are sensible to very small changes in data. XGBoost solves this problem by combining several decision trees (Chen and Guestrin, 2016). It starts by building a decision tree which finds a function $f_1(x)$ employing the input dataset x, whose output $\hat{y}_1$ resembles as much as possible the real output y. In the next iteration, a refined decision tree will fit the pseudo-residuals $f_2(x)$ from the first tree so that the result $\hat{y}_2$ approaches more to the output. This process is repeated for a user-defined number of iterations M, Eq. (2).

$$\begin{aligned} \hat{y}_1 &= f_1(x) \\ \hat{y}_2 &= \hat{y}_1 + f_2(x) \\ &\dots \\ \hat{y}_M &= \hat{y}_{M-1} + f_M(x) \end{aligned} \tag{2}$$

The new trees learn from previous trees' mistakes by increasing the weight of data points which were erroneously predicted. This way the

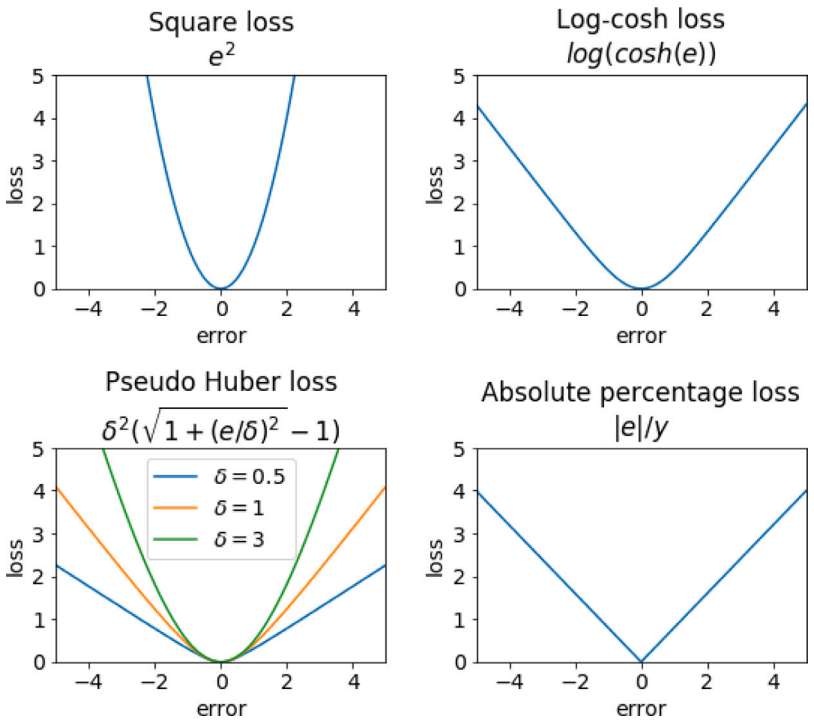

Fig. 4. Loss functions considered during the search for the most suitable one. e is the error between prediction $\hat{y}$ and real y values.

algorithm gives higher attention to large-error predictions. The rules describing how an algorithm penalizes errors are described by a loss function. The selected loss function should match the problem at hand and is crucial for the model outcomes.

XGBoost has been implemented in Python using the library *sklearn* (Pedregosa et al., 2011). The model was tuned for 20 rounds using *RandomizedSearchCV* with 3-fold cross validation, also available in *sklearn*. It is important to notice that performing cross-validation on all available data would cause data leakage (Bishop, 2006; Konstantinos Koutrombas, 2008). *Data leakage* refers to evaluating model performance on samples that the model has already seen, leading to over optimistic results. In order to prevent it, all the reported results were calculated on a left-out set (test set) being the remaining 20% of data. Cross validation was therefore performed on 80% of all the systems, hence 2/3 of this value was the training set (52.8% of all the data) and 1/3 (26.4% of all the data) was used as the validation set. *sklearn GroupShuffleSplit* function was employed to randomize the data so that the algorithm would learn the solar annual patterns. Squared error loss function together with RMSE as evaluation metric were used for model training. Outcome predictions were filtered to make all negative values equal to zero.

3.3. Loss functions

Contrary to utility scale PV forecasting, residential nowcasting requires high quality predictions for all individual systems. For this task the required loss function should be: (i) twice differentiable as required by XGBoost (Chen and Guestrin, 2016), (ii) sensitive to outliers and (iii) most importantly, relative to the size of the system. Thorough research and analysis was performed in search for the most suitable loss function for the problem at hand.

Square, log-cosh and pseudo Huber losses (vide Fig. 4) are the main options due to their differentiability (Wang et al., 2020). Amongst these, square loss is the most sensitive to outliers. Neither of the three is relative to the size of the system. A relative loss, such as the absolute percentage loss (Fig. 4) would fulfill this condition. However, it is not twice differentiable.

Several attempts to develop a custom loss function allowing to combine absolute and relative errors were made, but none of them succeeded. The main issue lies on the fact that loss functions focus by definition on larger errors than on smaller ones. Similar is the case for model utilizing Mean Absolute Percentage Error (MAPE) as training evaluation metric. Tackling this issue could be a starting point of future studies. Developing individual models for all systems was also considered, but it would not solve this issue, as the models would perform better in summer (higher yields) and worse in winter.

Overall, due to the failure of the aforementioned methods, suboptimal square loss function was used.

3.4. Metrics

This section describes the metrics employed to evaluate the performance of the model. We also aim to expose metric-related issues and present their impact on the quality of predictions.

Extensive overview of available metrics was provided by Antonanzas et al. (2016), Sobri et al. (2018), Zhang et al. (2015) who pursued both statistical and economic approaches. They described interesting metrics such as skew, curtosis, Renyi entropy and Kolmogorov–Smirnov Integral among others. However, the most popular metrics for solar yield nowcasting and forecasting still are RMSE, mean absolute error (MAE) and MAPE. MAE is the most intuitive, as it informs by how many kWh the predictions are off on average (Eq. (3)). Similarly, MAPE informs about average relative error in percentage terms (Eq. (4)), hence it is not affected by system size. RMSE also computes the average error although it additionally penalizes large errors (Eq. (5)). In these equations, $\hat{y}$ and y correspond to prediction and observation vectors respectively while n corresponds to the number of samples.

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i| \tag{3}$$

$$MAPE = \frac{1}{n}\sum_{i=1}^{n}\left|\frac{y_i - \hat{y}_i}{y_i}\right| \tag{4}$$

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2} \tag{5}$$

For this analysis, a metric that does not increase with system size, such as MAPE, could be the metric of choice. However, it has the drawback of taking high values for small values of yield. Other metrics used for this task are mean bias error (MBE, Eq. (6)), maximal error (max error) and skill score (SS, Eq. (7)). MBE allows negative and positive errors to cancel out, maximal error provides no information about error distribution and skill score is RMSE dependent hence influenced by the system size. Moreover, skill score informs about relative improvement with respect to persistence model which performance highly depends on the employed version of the model and on data resolution, as mentioned previously. These characteristics make the usage of skill score insufficient, even though it is promoted as a metric which allows comparison across projects (Yang, 2019).

$$MBE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i) \tag{6}$$

$$SS = 100 \cdot (1 - \frac{RMSE_{proposed}}{RMSE_{persistence}}) \tag{7}$$

An interesting approach was developed in Solar Monkey, where the absolute and relative error were combined in one *E-metric* described by Eqs. (8) and (9).

$$E_\epsilon = \frac{1}{n}\sum_{i=1}^{n} f(\hat{y}_i, y_i) \cdot 100\% \tag{8}$$

$$f(\hat{y}_i, y_i) = \begin{cases} 1 & \text{if } |\hat{y}_i - y_i| < \epsilon \\ 0 & \text{if } |\hat{y}_i - y_i| \geq \epsilon \end{cases} \tag{9}$$

Here, ϵ stands for the threshold of absolute error. Metric E_{10} gives a percentage of predictions with absolute error below 10 Wh. Similarly E_{50} gives a percentage of predictions with absolute error below 50 Wh and so on. The higher the E-metric is, the better the prediction. Its drawback is that it requires different thresholds for different data resolutions.

Another problem is that the majority of models is assessed on the entire test set without investigating results for individual systems. That is acceptable in case of forecasting performed for utility scale companies which operate the grid and do not need to know individual PV systems behavior. However, solar yield forecasting is becoming increasingly important for residential PV owners who would like to optimize their own production and consumption (Eneco, 2021). For instance, they could store solar energy and sell it to the grid during peak hours to maximize their profits. Moreover, early anomaly detection requires precise and accurate yield nowcasting for individual PV systems. General metric calculation might hide cases for which the model performs badly and does not include the fact that each system usually belongs to a different entity. In case of Solar Monkey, measuring overall model performance is informative, but insufficient to determine whether all customers receive predictions of high quality. Therefore, each of the previously presented metrics was calculated per system and stored in an array. Next, minimum, maximum, mean and standard deviation of each array were calculated. This approach allows also to identify the worst performing systems and narrows the scope of error analysis.

4. Results

This section presents the main results of the study showing the features selected by RFE, the performance of the proposed model compared with other models, and the learning curves and error analysis of XGBoost.

Table 1
Comparison of results calculated for all predictions together.

	R^2 [%]	MAE [kWh]	Max error [kWh]	RMSE [kWh]	MAPE [–]	SS [%]
Persistence	80	0.35	8.3	0.53	71	N/A
ElasticNet	83	0.30	32.2	0.47	11176	11
Polynomial Regression	90	0.24	30.1	0.37	10504	29
Random Forest	96	0.10	28.8	0.22	418	59
XGBoost	97	0.10	28.9	0.21	478	61

4.1. Feature selection

Features selected by RFE with their corresponding weights assigned by the XGBoost algorithm can be seen in Fig. 5. There, one can observe how the most influential feature is GHI followed by historical yields, which coincides with similar results obtained in literature (Isaksson and Conde, 2018; Abuella and Chowdhury, 2017). System size, one of the few system characteristics considered together with orientation, is also a relevant feature. Since our ML model was trained for on data from multiple PV systems, system size may be used as a scaling factor. The importance of cloud coverage increases as the forecast horizon decreases, being highly important for predictions less than 15 min ahead (Ahmed et al., 2020). Since the predictions are for the following hour, cloud coverage is important but does not make it to the top 3.

Even though their importance differ, all weather and descriptive features were kept by RFE. Out of all PV system parameters, only system size and orientation were selected, showing that most of the PV systems characteristics are not as influential as one may expect.

The non-measured features, namely the sun angles and day of year, have low relevance probably because the periodic information that they introduced is included in other features such as irradiance. Wind speed and precipitation are the two least relevant features, which coincides with previous results reported in literature (Abuella and Chowdhury, 2017; Kuzmiakova et al., 2017; Lee et al., 2019). One surprising result is the low importance of ambient temperature, which is usually amongst the chosen features (Abuella and Chowdhury, 2017; Kuzmiakova et al., 2017; Lee et al., 2019).

4.2. Models performance

After performing feature selection and hyperparameter tuning for XGBoost, the algorithm was trained to perform the forecasts. Since the objective is to monitor the residential PV systems, the lead time was instantaneous. The produced PV power was compared with the predicted one at the same instant of production to detect any malfunction. This occurred with an hourly resolution.

The algorithm's performance against the persistence method is presented in Tables 1 and 2. Next to XGBoost, performance of models based on ElasticNet, Polynomial Regression and Random Forest algorithms was presented for comparison. For information on these ML models, the reader is referred to Pedregosa et al. (2011). These models were selected for comparison not because they represent the state-of-the-art methods, but because they are easy to apply, provide fast results and can handle large amounts of data, hence they could be appropriate alternatives for a commercial application. Deep learning methods would probably obtain lower errors but their lack of interpretability and hard optimization would hinder their use in practice in the start-up.

XGBoost provides superior results with respect to all other analyzed models in most of the employed metrics. Random Forest has similar performance, with smaller maximum error and MAPE, however the higher RMSE implies that it makes predictions more often with large error. This is confirmed by the E-metrics in Table 2 where Random Forest has higher E_{10}, E_{50}, and E_{100}, but lower E_{500}. To increase trust in machine learning models one must make sure they do not make large

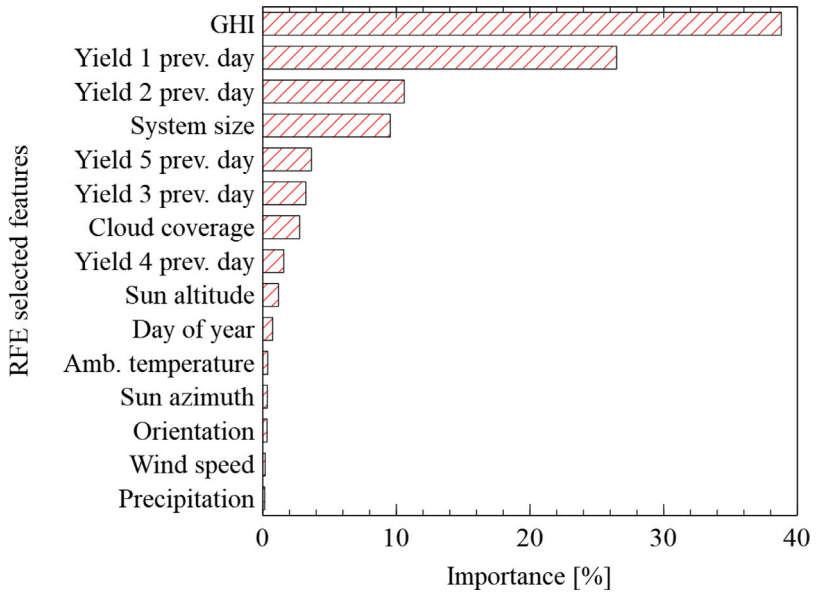

Fig. 5. Importance assigned by XGBoost to each of the features selected by RFE.

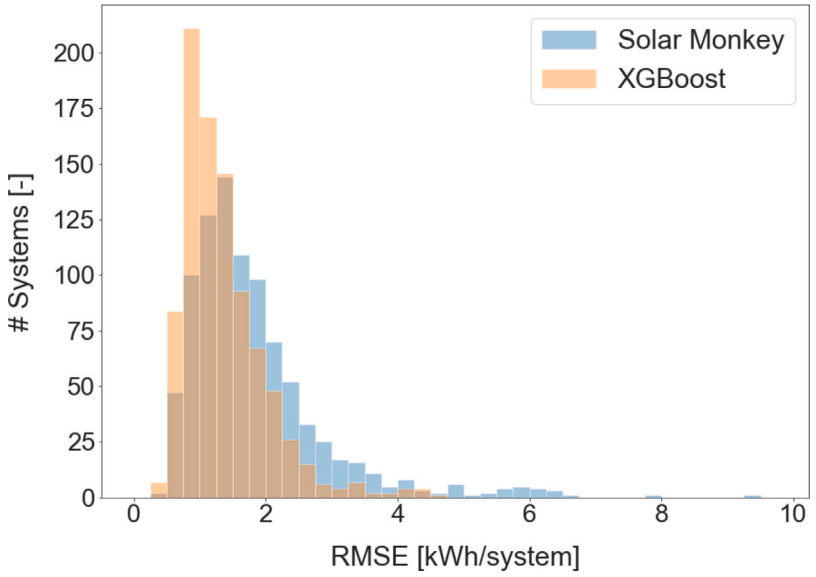

Fig. 6. Distribution of RMSE for individual systems comparing the performance of the commercial software with that of XGBoost.

Table 2
E-metrics for hourly rough data set.

	E_{10} [%]	E_{50} [%]	E_{100} [%]	E_{500} [%]
Persistence	11.0	22.1	32.9	75.2
ElasticNet	3.2	15.9	30.5	82.1
Polynomial Regression	4.2	20.5	38.2	87.8
Random Forest	27.4	57.4	72.5	96.4
XGBoost	23.6	54.7	71.4	96.7

errors. Therefore, E_{500} is the most important metric in this table, and is one of the reasons for choosing XGBoost over Random Forest.

Table 3 displays individual PV system metrics calculated for XGBoost and the analytical-based commercial software. It can be observed that XGBoost has individual system RMSE oscillating between 0.4 kWh and 4.6 kWh. The minimal RMSE of XGBoost is similar to the one of the analytical model while the maximal RMSE for XGBoost is around two times lower. Also, mean per-system RMSE decreased by around one third. Usage of the ML model caused a drop in relative error (mean MAPE) from around 44% to 23% which is almost two-fold improvement.

Distributions of per-system RMSE and per-system MAPE can be seen in Figs. 6 and 7 respectively. In both graphs, distributions corresponding to XGBoost model are shifted to the left with respect to the analytical model which indicates their higher quality.

4.3. Learning curves

Next to XGBoost metrics, learning curves are presented to depict the decrease in error as the model learns pattern in an increasingly large dataset. Learning curves provide several benefits. Firstly, they allow to

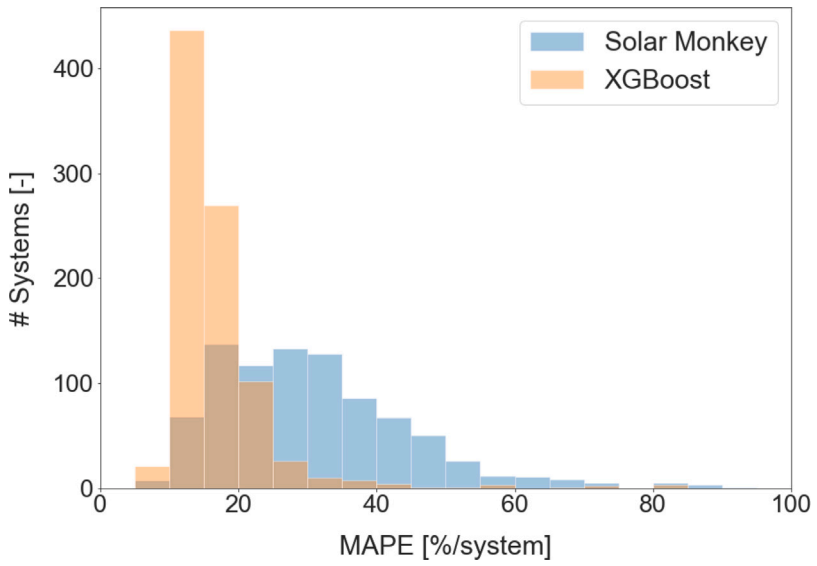

Fig. 7. Distribution of MAPE for individual systems comparing the performance of the commercial software with that of XGBoost.

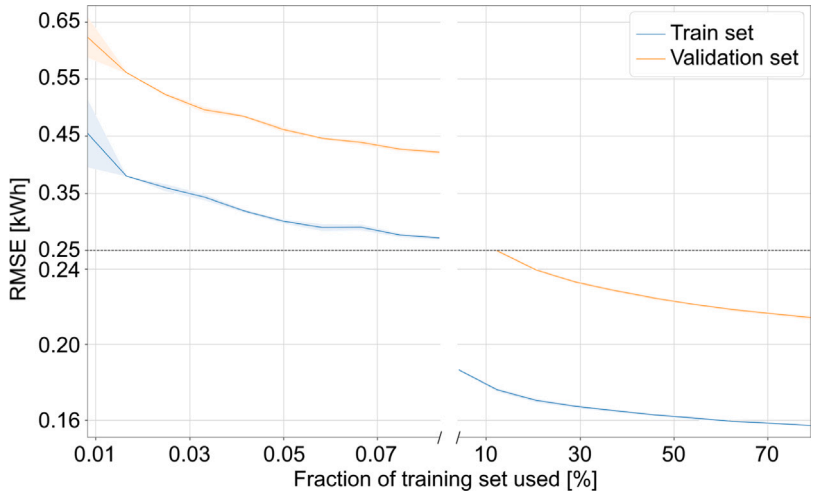

Fig. 8. XGBoost learning curve during and after the initial learning phase. Be aware of the change of scale in both axis. Curve plotted for custom training set sizes, with non-constant step value.

Table 3
Comparison of metrics for XGBoost vs. commercial software.

Individual system metrics	Solar Monkey	XGBoost
min RMSE [kWh]	0.37	0.40
max RMSE [kWh]	9.44	4.62
mean RMSE [kWh]	1.82	1.36
mean MAPE [%]	43.62	23.02

assess bias vs. variance trade-off, and make an informed decision about the next development step. Secondly, they allow to estimate the amount of data needed to decrease the error by a certain value, and therefore are important in financial calculations (data can be expensive). Finally, they allow to discover learning saturation — the point above which further training almost does not improve decrease error.

Learning curve in Fig. 8 was obtained using 3-fold cross-validation and allowed to determine whether full data set potential was utilized. Semi-transparent areas around the lines in the figure correspond to standard deviations of results for all three folds. It can be noticed that standard deviations are large for small data sets and decrease with increasing training set size. That is reasonable, as for large data sets it is less likely that particular shuffling of the training data would skew the result. Therefore, cross-validation can be neglected for data sets larger than 0.15% of the training set, that is exceeding 60,000 samples, as it significantly increases computational cost and has almost no impact on the results. Until seeing 0.15% of the training set size, XGBoost validation error decreases exponentially and it seems to stabilize later on. Drop in validation RMSE is only around 0.02 kWh for training set size between 40% and 80%. It can be concluded that XGBoost learning saturates around 40% of the training set, that is around 1.28 million samples.

In the analyzed data set, each system has 4,609 samples corresponding to non-zero irradiance in a year. Dividing critical number of samples (1.28 mln) by samples per system, it can be concluded that data for around 278 PV systems for an entire year is required for the XGBoost algorithm to reach a point beyond which the RMSE does not decrease significantly anymore. Further increase in training set size is likely to decrease error, but this small gain is not justified considering surge in required computational power and the associated financial cost. Other operations, e.g. data cleaning, are likely to provide larger gain in performance at lower expense. Usage of fewer number of systems with larger time horizon is likely to provide similar results, as long as the systems selected for training are representative of the test set.

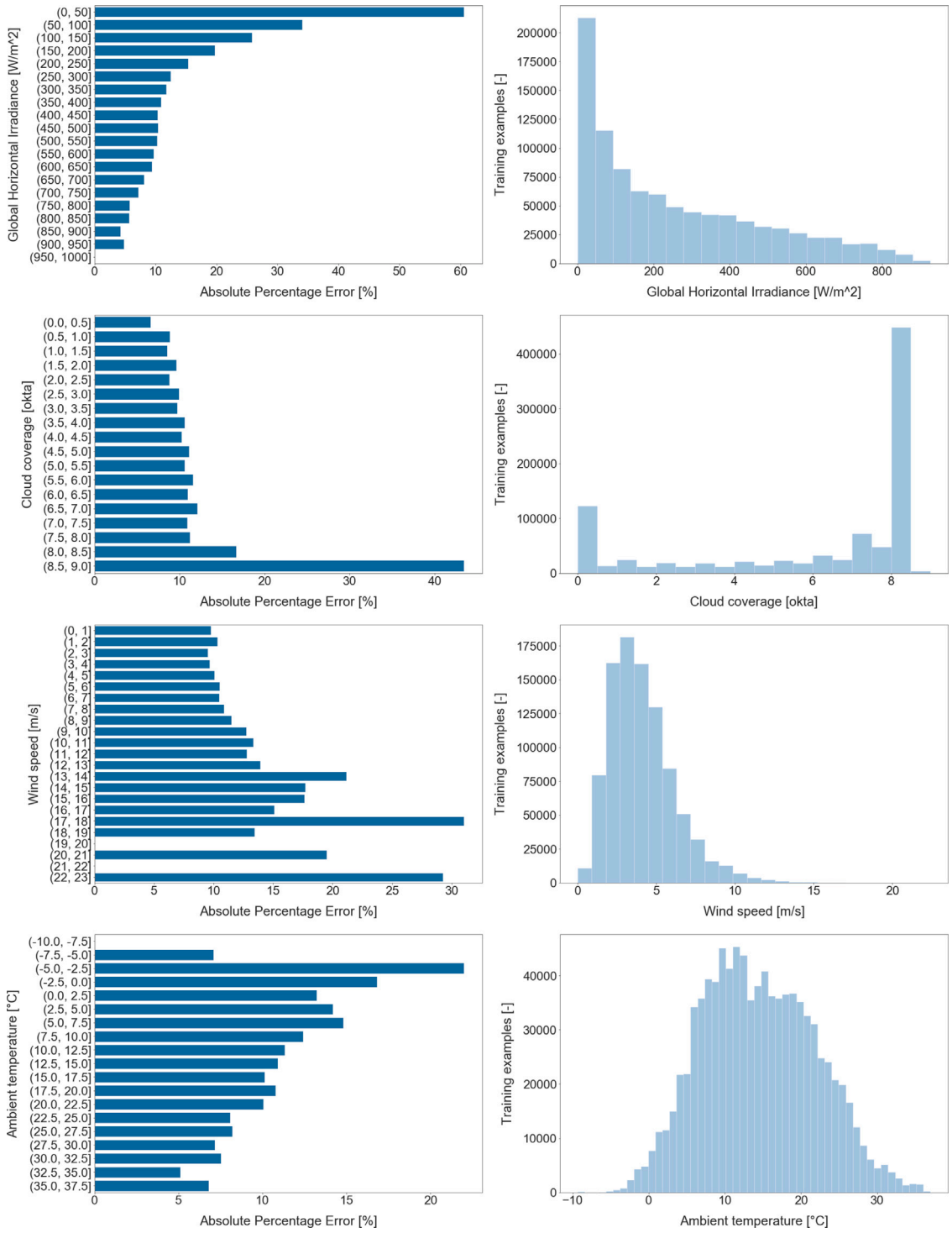

Fig. 9. Weather features as a function of absolute error.

4.4. Error analysis

This subsection describes when XGBoost model is mistaken and investigates the possible reasons of large errors. Based on literature, it is expected that the model works better under particular weather conditions, such as clear days (Rosiek et al., 2018) or seasons (Yin et al., 2020), or for certain systems (Gensler et al., 2017). This subsection focuses on finding the pattern in large XGBoost errors.

Fig. 9 depicts two graphs for each investigated parameter: distribution of absolute percentage error (on the left) and distribution of values (right hand side graphs). This was done to verify whether the magnitude of percentage error corresponds to the number of training samples in each bin. In Fig. 9 it can be seen that XGBoost's performance is independent of wind speed and ambient temperature, but depends heavily on GHI and to some extent on cloud coverage. For GHI and cloud coverage the largest absolute percentage errors overlap with bins containing the largest number of samples. This is not the case for wind

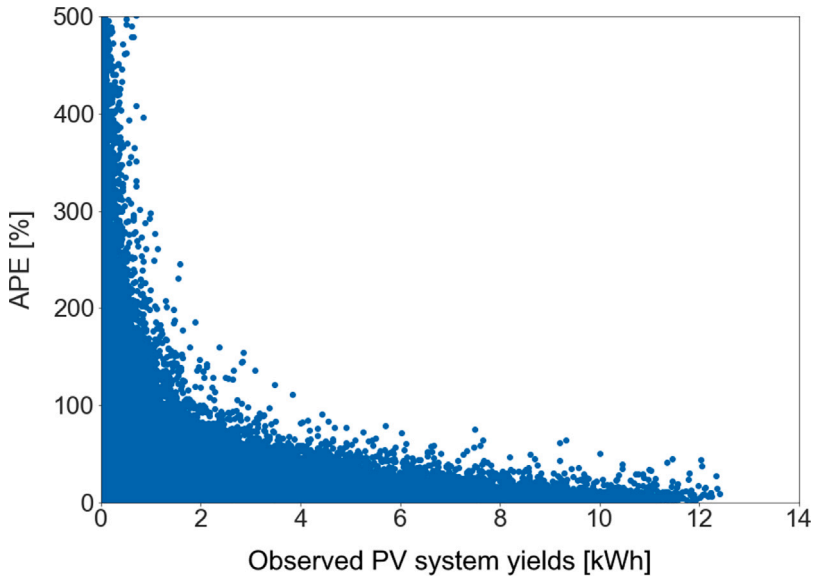

Fig. 10. Absolute percentage error as a function of observed yield for all systems.

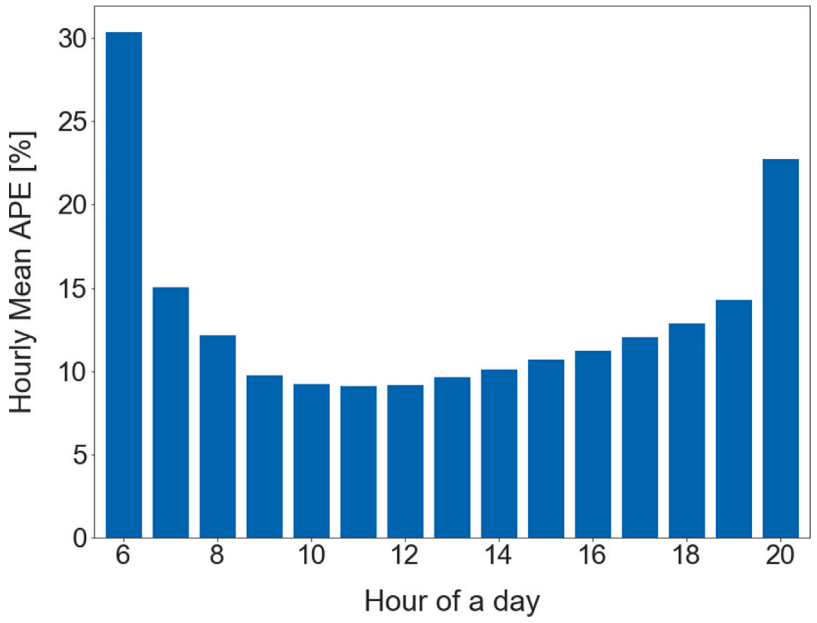

Fig. 11. Error analysis with respect to time of the day. Hours are expressed in local time.

speed and ambient temperature. As presented in Fig. 5 the model did not recognize these two features as relevant which means they have little influence on the final predictions, and therefore on the absolute percentage error.

Model's performance as a function of cloud coverage is presented in Table 4. It can be seen that the analyzed model performs around 50% better in case of clear sky conditions which indicates that modeling clouds influence on PV behavior is challenging. It also indicates that the utilized cloud coverage data is insufficient to precisely capture real life changes.

Next to the weather analysis, error analysis was performed. It was already described that utilization of squared error loss function favors large yields, that is large PV systems and sunny hours. To further verify this hypothesis, observed yields vs. their corresponding absolute percentage error (APE) were plotted in Fig. 10. It can be seen that APE has values far exceeding 100% for yield values below 2 kWh which confirms the initial assumption. APE with respect to hour of day and with respect to month were plotted in Figs. 11 and 12, respectively. Analysis of APE with respect to time in hours shows that relative error is the largest just after sunrise and just before sunset. Regarding the monthly analysis, it can be seen the model has the largest relative error for December followed by January. This is possibly due to these two months having the least irradiance in the whole year and therefore the smallest yields.

5. Conclusion

In this work we have shown an effective strategy for monitoring residential PV systems using ML techniques. XGBoost algorithm turned out to be the method of choice for the task of solar yield nowcasting on hourly data surpassing persistence, ElasticNet, Polynomial Regression, Random Forest and a commercially available analytical model. This work discussed that the currently used metrics and loss functions are insufficient for evaluation of solar nowcasting and forecasting models,

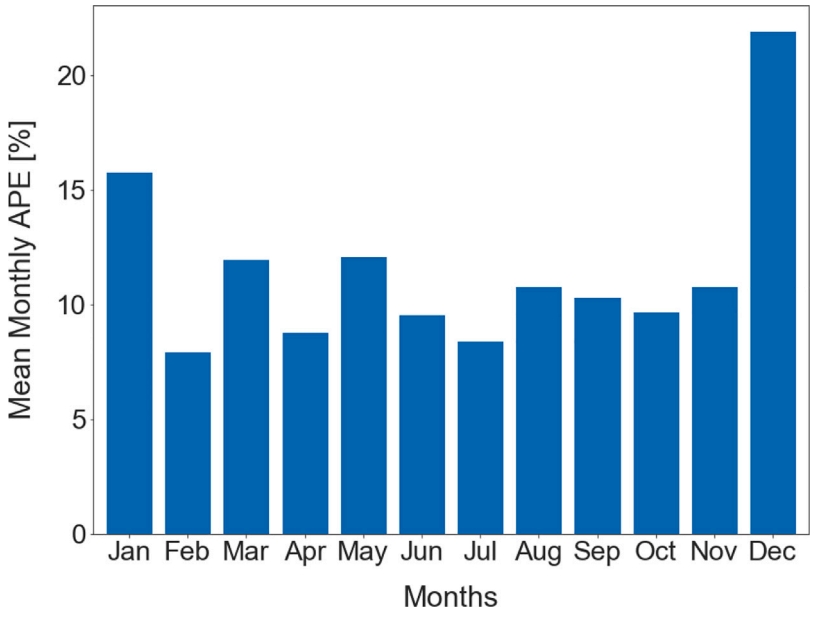

Fig. 12. Error analysis with respect to month of the year.

Table 4
XGBoost model performance with respect to cloud coverage.

	Metrics	Persistence	XGBoost
Clear sky (<1 okta)	MAE [kWh]	0.55	0.15
	RMSE [kWh]	0.73	0.28
	MAPE [%]	27	6
	R^2 [%]	69	95
Partly cloudy (1–7 okta)	MAE [kWh]	0.58	0.24
	RMSE [kWh]	0.77	0.38
	MAPE [%]	28	11
	R^2 [%]	60	90
Completely overcast (>7 okta)	MAE [kWh]	0.57	0.23
	RMSE [kWh]	0.76	0.37
	MAPE [%]	32	12
	R^2 [%]	46	87

as they fail to combine relative and absolute error. Combining the two is necessary for developing a single model able to learn individual system properties, independently of their size, and to obtain high quality residential scale solar yield predictions. In this study the usage of squared error loss function caused the model to be significantly mistaken for small values of yield occurring in winter, close to sunrise and sunset, and for small PV systems. Attempts to tackle this issue through sample normalization and using MAPE as training metric have failed. Further research should focus on developing new loss function and evaluation metrics. Despite the presented issues, XGBoost provides two fold improvement with respect to the commercially available analytical model.

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.

Acknowledgments

The research leading to these results has received funding from the Horizon 2020 Research and invention Programme, under Grant Agreement 952957, Trust-PV project. Special thanks should be given to Yves van Montfort who provided insight into inverter data quality issues, and to Eefje Visser who developed the E-metric. We would like to thank also Omkar Sane for valuable discussions about data analytics and machine learning used to conduct different parts of this study.

References

Abuella, M., Chowdhury, B., 2017. Solar Power Forecasting Using Support Vector Regression. Cornell University, arXiv:1703.09851.

Agoua, G., Girard, R., Kariniotakis, G., 2019. Probabilistic models for spatio-temporal. IEEE Trans. Sustain. Energy 10 (2), 780–789.

Ahmed, R., Sreeram, V., Mishra, Y., Arif, M.D., 2020. A review and evaluation of the state-of-the-art in PV solar power forecasting: Techniques and optimization. 124, 109792. http://dx.doi.org/10.1016/j.rser.2020.109792.

Akhter, M.N., Mekhilef, S., Mokhlis, H., Shah, N.M., 2019. Review on forecasting of photovoltaic power generation based on machine learning and metaheuristic techniques. 13, (7), pp. 1009–1023. http://dx.doi.org/10.1049/iet-rpg.2018.5649.

Anagnostos, D., Schmidt, T., Cavadias, S., Soudris, D., Poortmans, J., Catthoor, F., 2019. A method for detailed, short-term energy yield forecasting of photovoltaic installations. Renew. Energy 130, 122–129. http://dx.doi.org/10.1016/j.renene.2018.06.058.

Anon, 2021a. KNMI - koninklijk nederlands meteorologisch instituut. https://www.knmi.nl/home, Accessed: 2021-08-06.

Anon, 2021b. Solar monkey. https://solarmonkey.nl/, Accessed: 2021-08-06.

Antonanzas, J., Osorio, N., Escobar, R., Urraca, R., Martinez-de Pison, F.J., Antonanzas-Torres, F., 2016. Review of photovoltaic power forecasting. Sol. Energy 136, 78–111.

Bishop, C.M., 2006. Pattern Recognition and Machine Learning. In: Information Science and Statistics, Springer.

Brownlee, J., 2016. Feature importance and feature selection with xgboost in python. https://machinelearningmastery.com/feature-importance-and-feature-selection-with-xgboost-in-python/, Accessed: 25.11.2021.

Chen, T., Guestrin, C., 2016. Xgboost: A scalable tree boosting system. In: Proceedings of the 22nd Acm Sigkdd International Conference on Knowledge Discovery and Data Mining. pp. 785–794.

Das, U.K., Tey, K.S., Seyedmahmoudian, M., Mekhilef, S., Idris, M.Y.I., Van Deventer, W., Horan, B., Stojcevski, A., 2018. Forecasting of photovoltaic power generation and model optimization: A review. 81, pp. 912–928. http://dx.doi.org/10.1016/j.rser.2017.08.017.

De Felice, M., Petitta, M., Ruti, P.M., 2015. Short-term predictability of photovoltaic production over Italy. Renew. Energy 80, 197–204. http://dx.doi.org/10.1016/j.renene.2015.02.010, arXiv:1409.8202.

de Vries, T.N., Bronkhorst, J., Vermeer, M., Donker, J.C., Briels, S.A., Ziar, H., Zeman, M., Isabella, O., 2020. A quick-scan method to assess photovoltaic rooftop potential based on aerial imagery and LiDAR. Sol. Energy 209 (February), 96–107. http://dx.doi.org/10.1016/j.solener.2020.07.035.

Elsinga, B., van Sark, W.G., 2017. Short-term peer-to-peer solar forecasting in a network of photovoltaic systems. Appl. Energy 206, 1464–1483.

Eneco, 2021. Our exits: peeeks. https://www.eneco.com/wat-we-doen/Ventures/onze-exits/, Accessed: 2021-08-09.

Ferlito, S., Adinolfi, G., Graditi, G., 2017. Comparative analysis of data-driven methods online and offline trained to the forecasting of grid-connected photovoltaic plant production. Appl. Energy 205, 116–129. http://dx.doi.org/10.1016/j.apenergy.2017.07.124.

Fu, L., Yang, Y., Yao, X., Jiao, X., Zhu, T., 2019. A regional photovoltaic output prediction method based on hierarchical clustering and the mRMR criterion. Energies 12 (20), http://dx.doi.org/10.3390/en12203817, URL https://www.mdpi.com/1996-1073/12/20/3817.

Gensler, A., Henze, J., Sick, B., Raabe, N., 2017. Deep Learning for solar power forecasting - An approach using AutoEncoder and LSTM Neural Networks. In: 2016 IEEE International Conference on Systems, Man, and Cybernetics, SMC 2016 - Conference Proceedings. pp. 2858–2865. http://dx.doi.org/10.1109/SMC.2016.7844673.

Granitto, P.M., Furlanello, C., Biasioli, F., Gasperi, F., 2006. Recursive feature elimination with random forest for PTR-MS analysis of agroindustrial products. Chemometr. Intell. Lab. Syst. 83 (2), 83–90.

Guyon, I., Weston, J., Barnhill, S., Vapnik, V., 2002. Gene selection for cancer classification using support vector machines. Mach. Learn. 46 (1), 389–422.

Han, S., hui Qiao, Y., Yan, J., qian Liu, Y., Li, L., Wang, Z., 2019. Mid-to-long term wind and photovoltaic power generation prediction based on copula function and long short term memory network. Appl. Energy 239, 181–191. http://dx.doi.org/10.1016/J.APENERGY.2019.01.193.

Holmgren, W.F., Hansen, C.W., Mikofski, M.A., 2018. Pvlib python: a python package for modeling solar energy systems. J. Open Source Softw. 3 (29), 884. http://dx.doi.org/10.21105/joss.00884.

Hong, T., Pinson, P., Fan, S., Zareipour, H., Troccoli, A., Hyndman, R.J., 2016. Probabilistic energy forecasting: Global Energy Forecasting Competition 2014 and beyond. 32, (3), pp. 896–913. http://dx.doi.org/10.1016/j.ijforecast.2016.02.001.

Hossain, M., Mekhilef, S., Danesh, M., Olatomiwa, L., Shamshirband, S., 2017. Application of extreme learning machine for short term output power forecasting of three grid-connected PV systems. J. Clean. Prod. 167, 395–405. http://dx.doi.org/10.1016/j.jclepro.2017.08.081.

Isaksson, E., Conde, M.K., 2018. Solar power forecasting with machine learning techniques. Tech. rep., KTH Royal Institute of Technology.

Jung, Y., Jung, J., Kim, B., Han, S.U., 2020. Long short-term memory recurrent neural network for modeling temporal patterns in long-term power forecasting for solar PV facilities: Case study of South Korea. J. Clean. Prod. 250, 119476. http://dx.doi.org/10.1016/J.JCLEPRO.2019.119476.

Kleissl, J., 2013. Solar Energy Forecasting and Resource Assessment. Academic Press.

Konstantinos Koutrombas, S.T., 2008. Pattern Recognition. Elsevier.

Kuzmiakova, A., Colas, G., McKeehan, A., 2017. Short-term Memory Solar Energy Forecasting at University of Illinois. (December), University of Illinois, pp. 1–6.

Lee, D., Jeong, J., Yoon, S.H., Chae, Y.T., 2019. Improvement of short-term BIPV power predictions using feature engineering and a recurrent neural network. Energies 12 (17), 3247. http://dx.doi.org/10.3390/EN12173247, https://www.mdpi.com/1996-1073/12/17/3247.

Lee, W., Kim, K., Park, J., Kim, J., Kim, Y., 2018. Forecasting solar power using long-short term memory and convolutional neural networks. IEEE Access 6, 73068–73080. http://dx.doi.org/10.1109/ACCESS.2018.2883330.

Li, Y., He, Y., Su, Y., Shu, L., 2016. Forecasting the daily power output of a grid-connected photovoltaic system based on multivariate adaptive regression splines. Appl. Energy 180, 392–401. http://dx.doi.org/10.1016/j.apenergy.2016.07.052.

Lin, K.P., Pai, P.F., 2016a. Solar power output forecasting using evolutionary seasonal decomposition least-square support vector regression. J. Clean. Prod. 134 (Part B), 456–462. http://dx.doi.org/10.1016/j.jclepro.2015.08.099.

Lin, K.P., Pai, P.F., 2016b. Solar power output forecasting using evolutionary seasonal decomposition least-square support vector regression. J. Clean. Prod. 134 (Part B), 456–462. http://dx.doi.org/10.1016/j.jclepro.2015.08.099.

Loh, W.Y., 2011. Classification and regression trees. Wiley Interdiscip. Rev.: Data Min. Knowl. Discov. 1 (1), 14–23. http://dx.doi.org/10.1002/widm.8.

Lorenz, E., Scheidsteger, T., Hurka, J., Heinemann, D., Kurz, C., 2011. Regional PV power prediction for improved grid integration. Prog. Photovolt., Res. Appl. 19 (7), 757–771.

Maitanova, N., Telle, J.-S., Hanke, B., Schmidt, T., Grottke, M., von Maydell, K., Agert, C., 2020. Machine learning approach to a low-cost day-ahead photovoltaic power prediction based on publicly available weather reports. Energies 13 (3), http://dx.doi.org/10.3390/en13030735.

Massaoudi, M., Chihi, I., Sidhom, L., Trabelsi, M., Refaat, S.S., Oueslati, F.S., 2019. PV power forecasting using weighted features for enhanced ensemble method. IEEE Access arXiv:1910.09404, URL http://arxiv.org/abs/1910.09404.

Ni, Q., Zhuang, S., Sheng, H., Kang, G., Xiao, J., 2017. An ensemble prediction intervals approach for short-term PV power forecasting. Sol. Energy 155, 1072–1083. http://dx.doi.org/10.1016/j.solener.2017.07.052.

Notton, G., Nivet, M.L., Voyant, C., Paoli, C., Darras, C., Motte, F., Fouilloy, A., 2018. Intermittent and stochastic character of renewable energy sources: Consequences, cost of intermittence and benefit of forecasting. 87, pp. 96–105. http://dx.doi.org/10.1016/j.rser.2018.02.007.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., Duchesnay, E., 2011. Scikit-learn: Machine learning in python. J. Mach. Learn. Res. 12, 2825–2830.

Philipps, D.S., Fraunhofer, I., Warmuth, W., 2021. Photovoltaics report. Tech. rep., Fraunhofer Institute for Solar Energy Systems, Freiburg, Germany.

Photovoltaic Power Systems Program, 2021. Snapshot of global PV markets. Tech. rep., International Energy Agency, URL http://www.iea-pvps.org.

Pierro, M., De Felice, M., Maggioni, E., Moser, D., Perotto, A., Spada, F., Cornaro, C., 2017. Data-driven upscaling methods for regional photovoltaic power estimation and forecast using satellite and numerical weather prediction data. Sol. Energy 158, 1026–1038. http://dx.doi.org/10.1016/j.solener.2017.09.068.

Rosiek, S., Alonso-Montesinos, J., Batlles, F.J., 2018. Online 3-h forecasting of the power output from a BIPV system using satellite observations and ANN. Int. J. Electr. Power Energy Syst. 99, 261–272. http://dx.doi.org/10.1016/j.ijepes.2018.01.025.

Sobri, S., Koohi-Kamali, S., Rahim, N.A., 2018. Solar photovoltaic generation forecasting methods: A review. Energy Convers. Manage. 156, 459–497.

Sun, Y., Venugopal, V., Brandt, A.R., 2019a. Short-term solar power forecast with deep learning: Exploring optimal input and output configuration. Sol. Energy 188, 730–741. http://dx.doi.org/10.1016/j.solener.2019.06.041.

Sun, Y., Venugopal, V., Brandt, A.R., 2019b. Short-term solar power forecast with deep learning: Exploring optimal input and output configuration. Sol. Energy 188, 730–741. http://dx.doi.org/10.1016/j.solener.2019.06.041.

Theocharides, S., Makrides, G., Georghiou, G.E., Kyprianou, A., 2018. Machine learning algorithms for photovoltaic system power output prediction. In: 2018 IEEE International Energy Conference. ENERGYCON, IEEE, pp. 1–6.

Vorhies, W., 2016. Want to win competitions? Pay attention to your ensembles. URL https://www.datasciencecentral.com/profiles/blogs/want-to-win-at-kaggle-pay-attention-to-your-ensembles.

Wang, Q., Ma, Y., Zhao, K., Tian, Y., 2020. A comprehensive survey of loss functions in machine learning. Ann. Data Sci. 1–26.

Wang, K., Qi, X., Liu, H., 2019. Photovoltaic power forecasting based LSTM-Convolutional Network. Energy 189, 116225. http://dx.doi.org/10.1016/J.ENERGY.2019.116225.

Wolff, B., Kühnert, J., Lorenz, E., Kramer, O., Heinemann, D., 2016. Comparing support vector regression for PV power forecasting to a physical modeling approach using measurement, numerical weather prediction, and cloud motion data. Sol. Energy 135, 197–208. http://dx.doi.org/10.1016/j.solener.2016.05.051.

Yang, D., 2019. A guideline to solar forecasting research practice: Reproducible, operational, probabilistic or physically-based, ensemble, and skill (ROPES). 11, (2), p. 22701. http://dx.doi.org/10.1063/1.5087462.

Yin, W., Han, Y., Zhou, H., Ma, M., Li, L., Zhu, H., 2020. A novel non-iterative correction method for short-term photovoltaic power forecasting. Renew. Energy 159, 23–32. http://dx.doi.org/10.1016/J.RENENE.2020.05.134.

Zhang, J., Florita, A., Hodge, B.-M., Lu, S., Hamann, H.F., Banunarayanan, V., Brockway, A.M., 2015. A suite of metrics for assessing the performance of solar power forecasting. Sol. Energy 111, 157–175.

Zhang, W., Li, Q., He, Q., 2022. Application of machine learning methods in photovoltaic output power prediction: A review. J. Renew. Sustain. Energy 14 (2), 022701. http://dx.doi.org/10.1063/5.0082629, URL https://aip.scitation.org/doi/abs/10.1063/5.0082629.