Article

Leveraging Explainable Artificial Intelligence in Solar Photovoltaic Mappings: Model Explanations and Feature Selection

Eduardo Gomes [1,2,3], Augusto Esteves [1,3], Hugo Morais [1,2,*] and Lucas Pereira [1,3]

1 Instituto Superior Técnico—IST, Universidade de Lisboa, 1749-016 Lisboa, Portugal; eduardo.c.gomes@tecnico.ulisboa.pt (E.G.); augusto.esteves@tecnico.ulisboa.pt (A.E.); lucas.pereira@tecnico.ulisboa.pt (L.P.)
2 INESC-ID—Instituto de Engenharia de Sistemas e Computadores-Investigação e Desenvolvimento, 1000-029 Lisboa, Portugal
3 ITI/LARSyS—Interactive Technologies Institute, 1900-319 Lisboa, Portugal
* Correspondence: hugo.morais@tecnico.ulisboa.pt

Abstract: This work explores the effectiveness of explainable artificial intelligence in mapping solar photovoltaic power outputs based on weather data, focusing on short-term mappings. We analyzed the impact values provided by the Shapley additive explanation method when applied to two algorithms designed for tabular data—XGBoost and TabNet—and conducted a comprehensive evaluation of the overall model and across seasons. Our findings revealed that the impact of selected features remained relatively consistent throughout the year, underscoring their uniformity across seasons. Additionally, we propose a feature selection methodology utilizing the explanation values to produce more efficient models, by reducing data requirements while maintaining performance within a threshold of the original model. The effectiveness of the proposed methodology was demonstrated through its application to a residential dataset in Madeira, Portugal, augmented with weather data sourced from SolCast.

Keywords: explainable artificial intelligence; feature selection; machine learning; photovoltaic seasonality

1. Introduction

Solar Photovoltaic (PV) systems are being widely deployed, transforming the global energy landscape by harnessing abundant solar power. This proliferation offers several benefits, including reduced carbon emissions, enhanced energy independence, and lower electricity costs. However, it also presents challenges, such as managing the inconsistency of solar output and the growing need for energy storage solutions to ensure reliable supply [1].

Hence, accurate forecasting of PV production has become crucial to the global energy transformation. For instance, accurate forecasts help to balance supply and demand, preventing issues like overloading or underutilization [2,3]. In energy communities, forecasts enable members to optimize shared resources, plan energy usage, and make informed decisions about trading energy [4]. Additionally, PV forecasts support better energy storage management, ensuring that batteries are charged and discharged efficiently [5], and aid in scheduling the charging of electric vehicles at optimal times [6].

With the increased use of Machine Learning (ML) and the development of increasingly intricate models, uncovering the reasoning behind model behavior and associated decision-making processes becomes progressively more challenging. Use of eXplainable Artificial

Intelligence (XAI) is gaining popularity as a potential solution to these challenges [7,8]. Specifically, DARPA's XAI program emphasizes the need for artificial intelligence systems to be interpretable, to ensure transparency, and build user trust in various domains such as medicine and finance [7]. DARPA's work also highlighted the role and impact of explanations for artificial intelligence models. XAI methodologies were approached in the context of energy and power systems in [8], to enhance model reliability and build user confidence that decisions are correct and logical. Additionally, methods such as Local Interpretable Model-Agnostic Explanation (LIME) and SHapley Additive exPlanations (SHAP) were discussed and evaluated by their effectiveness. The use of XAI provides further information to the user and justifies the model outputs, allowing greater accuracy and transparency regarding decisions. Depending on the target audience, XAI may prove useful in avoiding model bias and unreasonable decisions. The potential for XAI applications in this domain remains untapped and could be leveraged to provide better selection mechanisms, as an understanding of the relationships between performance and feature importance could potentially benefit forecast approaches.

In this regard, XAI has emerged as a valuable tool in PV forecasting, particularly for complex models such as neural networks. Studies have highlighted the importance of understanding the relationships between inputs and outputs, particularly for key features like solar irradiance, temperature, and humidity [9–11]. For instance, the authors of [10] used XAI to study the relationships between features of the dataset used, in addition to data analysis, error estimation, and fault detection. Another instance of using XAI for uncovering relationships between features can be found in [11], where environmental features were studied concerning model outputs. Similarly, the authors of [12] drew attention to an uncovered relationship between environmental features and PV output, with higher altitudes, in particular, as the most important feature driving the production, constituting one of their key findings and takeaway recommendations. Fault detection in PV systems has been studied employing different XAI strategies [13], with SHAP demonstrating increased stability and consistency over other studied methods. However, the literature has not explored the variation in such relationships across seasons. Furthermore, the possibility of using XAI for feature selections has only briefly been touched upon by the research community, with mixed results: one study reported improved accuracy after removing less important features, while another found a decline in performance [14,15].

Against this background, this paper presents two original research contributions in the field of PV interpretability and XAI, specifically

- We present a methodological framework for applying XAI to PV forecasting algorithms using SHAP local explanations. This framework was applied to a real-world dataset comprising two years of PV forecasting data from Madeira, Portugal, to uncover how feature importance varied throughout the year. By focusing on seasonal variations, this research offers valuable insights into how different factors influence PV forecasting models over time, addressing a gap in the literature where XAI's application across seasons has been largely underexplored. The results showed an improvement in the performance of the forecasting methods (accuracy and computational cost).
- A feature selection methodology for PV forecasting, based on SHAP techniques, aimed at reducing computational costs through an informed reduction in the feature space and consequently of the model size. This methodology was evaluated using the dataset mentioned in the previous contribution. A comparison with classic feature extraction methods, namely *Spearman* correlation and variance threshold, is also presented.

The structure of this paper is as follows: Section 2 presents some background and related works on the topics of PV forecasting, XAI, and their intersection. Section 3 presents

the methods proposed in this paper, namely the development of model explanations and feature selection leveraging SHAP values. Section 4 describes the evaluation procedure, including an explanation of the datasets and the two experiments that were designed to assess the validity of the proposed methods. The results and discussion are presented in Section 5. The paper concludes in Section 6, with a summary of the main findings, limitations, and suggestions for future research.

2. Background and Related Works

2.1. PV Production Forecasting

Despite advances in PV production forecasting and the increasing availability of historical data, it remains an active research topic due to dependencies on variables such as geographic location, weather patterns, desired forecast horizons, and other phenomena. Depending on the variables used, model generalization and training can be quite complex, with works tackling short-term forecasting to long-term forecasting, as well as presenting a myriad of different algorithms, ranging from simple linear regressions to ensemble methods and, more recently, Deep Learning (DL) [16–19].

Previous works and systematic reviews mentioned the importance of studying input relationships concerning outputs and applying methods such as correlations to identify and select variables that should be of more importance and help model learning. One such feature is solar irradiance, with several mentions of cloud opacity and module temperature [20]. In [18], the authors summarized the findings of other approaches and the data used to perform PV forecasting, and presented their choice of features through the aforementioned correlation, namely solar irradiance, temperature, and humidity.

Forecasting performance can result in direct economic consequences, where this information can influence usage of energy generation reserves and peer-to-peer energy trading markets. Therefore, accurate PV forecasts are in demand, and a lack of accuracy can lead to loss of revenue [21,22]. Similarly, accurate forecasts have been studied in the context of energy markets, with smaller errors benefiting the users and leading to increased profits [23].

Several works have previously investigated the impact of seasonality in the context of PV production [24–26]. In [24], the authors reported that machine learning models that consider seasonality tend to exhibit lower error rates, between 5% and 25%, compared to models that do not consider such variables. Works such as [25] have attempted to use and predict seasonal traits that may benefit more accurate outputs.

2.2. eXplainable Artificial Intelligence (XAI)

XAI is an area of research propelled by the question, "Why do models output the values they do?" Some of its main contributions to the AI community include building trust in trained models, model debugging, model fine-tuning, and reasoning [27,28]. This is achieved by calculating and assigning impact weights for features, such as specific pixel values for areas of an image or an overall weight for a variable in a regression problem.

While some ML algorithms are very simple to explain and it is easy to infer the reasons for outputs and model behavior (e.g., linear regressions and simple decision trees), other approaches, such as neural networks, work in ways that are not as easily understood. As there is usually a trade-off between model performance and explainability, XAI is advantageous as it can provide insights into these black-box models, where the behavior is not always apparent and there is not a straightforward explanation of the outputs [27,29,30]. This helps to build model trust for the end-users, as explanations may validate the model reasoning for the users [27]. For example, a community manager may validate model decisions taken by a model when its behavior is more explicit and justified.

Methods such as Grad-CAM and Integrated Gradients have emerged and use model internals (i.e., neural network layer gradients). Grad-CAM can highlight the pixel importance in images and provide a visual representation for the end-user as to why the model provided its output [31]. However, model internals are not always available. Therefore, methods such as LIME and SHAP have been proposed [32,33]. These methods focus on varying the model input and studying model outputs, establishing relationships between features and calculating their respective importance regarding the outputs. In XAI, there is also a distinction between local and global explainability. Local explainability aims to provide insights into the reasoning behind individual prediction values. In contrast, global explainability focuses on understanding the model's overall behavior and decision-making process.

2.3. PV Forecasting and XAI

While the importance of certain features has been well established in the literature, their impact on PV forecasting is not always apparent [16]. XAI can thus be considered and applied to PV forecasts to understand how the features are directly related to outputs.

Some works developed in this area have used XAI to gain insights into how a model makes its predictions [14,15]. These works applied XAI to random forests and XGBoost ensemble algorithms, respectively. In [14], the authors provided a comparison between LIME, SHAP, and Explain Like I'm 5 (ELI5), and reported that SHAP was the only method among the ones used that delivered a global explanation, although being the method that was computationally more expensive and time-consuming. In [15], the authors used ELI5 and reported Root Mean Squared Error (RMSE) scores for models built with all available features and models built with just a subset, showing a decline in performance for the latter. Both works highlight the need for explainable models to improve efficiency and trust in forecast solutions.

In [9], the authors explored the application of SHAP to probabilistic models in the context of PV forecasting, comprehensively analyzing model behavior learned feature importance and conducting experiments across seasons. However, this work did not examine how SHAP values varied across seasons. Given that factors like solar irradiance, temperature, and weather patterns change significantly with the seasons, understanding these variations in SHAP values could offer deeper insights into feature impacts and lead to more accurate and robust forecasting.

Utilizing XAI as a feature selection mechanism has only seldom been reported in the literature [9,15], and no clear methodology to achieve this has been provided. For example, in [9], the authors experimented with removing weather features such as information on precipitation, temperature, and wind speed, since these were not consistently used across the models. A decrease of around 6% in the RMSE metric was observed when discarding these features, indicating that using explanation as a feature selection can be a promising approach. In [15], a similar experiment was developed, where the authors trained a second model after removing the two least important features according to the ELI5 method. Interestingly, in this case, the authors reported an increase in the RMSE in the second model (from 7.22 with all features to 8.22 with a subset of features). Ultimately, these findings highlight the need for further research to better understand when and how XAI can effectively guide feature selection.

2.4. Summary

Overall, this short literature review underscores the importance of XAI in PV production forecasting, particularly in understanding complex models and building trust through transparent predictions. Seasonality is critical, with models that account for seasonal varia-

tions generally showing improved accuracy. Furthermore, while XAI offers the potential
to guide feature selection to enhance forecasting, results have been mixed, indicating the
need for further research to establish reliable methodologies.

3. Methods

3.1. Model Explanations with SHAP

The methodology for explaining models using SHAP values is depicted in Figure 1.
The first step involves training and testing different PV production mapping algorithms,
including a hyperparameter search to optimize the model outputs. Then, the trained model
and the test datasets are used to calculate and rank the impact of the different features
according to their SHAP value.
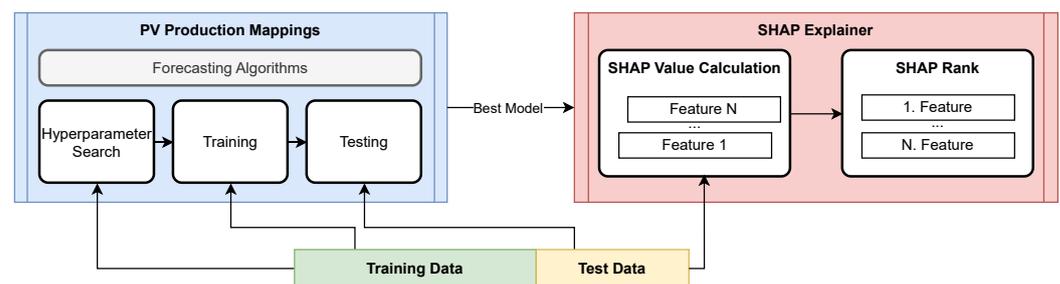
Figure 1. Proposed methodology for explaining PV production mappings using SHAP values.

3.1.1. PV Production Mapping Algorithms

The work presented in this paper leverages two ML algorithms, XGBoost [34] and Tab-
Net [35]. These algorithms were selected for being fundamentally distinct, with XGBoost
being a decision tree ensemble and TabNet being a Deep Neural Network (DNN) architec-
ture, as well as not having applications limited to time series data, unlike approaches such
as ARIMA and recurrent neural networks.

XGBoost

The XGBoost algorithm is a modified boosted decision tree ensemble with several
optimizations for better scalability. One of the adopted mechanisms for optimizing the
algorithm lies in a function approximation for splitting features instead of relying on the
exact greedy algorithm, making the process more efficient. It was also developed to tackle
sparse data, i.e., missing rows in data, contributing to the model's robustness and further
improving model training times. This algorithm also features an additional regularization
term in its loss function, which helps prevent overfitting, with the authors mentioning that,
in practice, it also affects model complexity, favoring less complex models. Other factors
contributing to its adoption are the training and inference speeds and consistency between
runs. These factors allow for better model prototyping, speed up the process, and ensure
the results are reproducible within an acceptable margin. In this work, the packages *xgboost*
(1.7.6), *tensorflow* (2.10.0), *tabnet* (0.1.6), and *shap* (0.42.1) were used.

TabNet

The TabNet algorithm, developed by Google, aims to harness the power of deep neural
networks (DNNs) to create effective models for tabular data, a domain where DNNs are
not as dominant compared to fields like computer vision [35]. The TabNet architecture is
built around two core components: Attention Transformer Blocks and Feature Transformer
Blocks. These blocks are sequentially stacked, with each additional block increasing the
model size and complexity. Feature Transformer Blocks process input features to produce
attention and feature output vectors, with the attention output passed to the subsequent

Attention Transformer. The Attention Transformer layers are designed to extract the significance of features, allowing for the internal selection of relevant features for each sample. In this work, we used an implementation of the TabNet found on GitHub (TabNet, https://github.com/titu1994/tf-TabNet (accessed on 4 of November 2024) (version: 0.1.6) that provided support for TensorFlow 2.0 versions).

3.1.2. Training and Testing Procedures

Before model training, a hyperparameter search was conducted using Optuna [36] to search the parameter space more efficiently, set to 100 trials, with each trial running for 20 epochs and reporting the RMSE on the test set. This allowed the model to define parameters to learn and generalize better on a given task. The hyperparameters and their search space are presented in Table 1. The search space for TabNet was initially defined as suggested by the original paper [35], although empirical knowledge allowed narrowing down variables and their respective values. The computational overheads in the hyperparameter optimization (HPO) step varied according to the chosen algorithm, with XGBoost being faster and TabNet comparatively more time-consuming. However, this overhead applies only to the HPO step and did not affect the subsequent model training and inference. The code used for the hyperparameter optimization can be found on GitHub (Code for HPO https://github.com/ECGomes/pv_forecast/blob/main/model_hyperparams.py (accessed on 5 of March 2025)).

Table 1. Hyperparameter search space.

XGBoost	TabNet
Number of Estimators (50–5000)	Feature Dimension Space (32–72)
Maximum Depth (10–5000)	Output Dimension Space (4–12)
Alpha regularizer (10^{-5}–10^{-3})	Number of Decision Steps (1–4)
Lambda regularizer (10^{-5}–10^{-3})	

After the hyperparameter search, ten models were trained with random seeds for each algorithm, to minimize variance across runs, and the best-performing model was selected. The selection criterion was the daily average RMSE, a metric commonly used in related works (e.g., [9,15]). Specifically, the RMSE was calculated for each day in the test set and then averaged across all days. The RMSE is defined in Equation (1), where n represents the number of data records considered, and y and $\hat{y}$ stand for the ground truth and predicted values, respectively .

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

The XGBoost models ran until convergence, while the TabNet models had an EarlyStopping callback that allowed halting the training once a certain amount of epochs had passed and the validation score had not improved , set to 20 epochs. Each model had as input a single record of data containing all the features described above for the dataset. The output of both models was a single point with the prediction of solar production, i.e., feature values at noon to output PV values at that same instant.

The models were trained using TensorFlow on a machine configured with an NVIDIA 3070, AMD Ryzen 9 5900X, and 128 Gb RAM [37]. The seed values for *Python* (3.9), *Numpy* (1.24.4), *Tensorflow* (2.10.0), and *Optuna* (3.3.0) were set to 69 for consistency and to allow reproducibility.

3.1.3. SHAP Explainer

In this work, we used SHAP values [33], a model-agnostic method that can be applied to pretrained models. SHAP was also selected for its higher consistency [13] and strong theoretical foundation, as well as being able to provide local and global explanations [33]. The values produced and presented in this work range from negative to positive. In this context, negative SHAP can be interpreted as influencing the model to output lower forecast values. Similar reasoning can be applied to positive SHAP, promoting higher forecast values. It is important to mention that, while an impact may be positive, it may occur when a feature takes a negative value, with the reverse also being true. An example of this would be higher temperature values decreasing cell performance and thus hurting the forecasts.

To achieve generalization, the SHAP explainer was employed for the entirety of the test set, obtaining the SHAP values for each feature for each sample in the dataset. To rank features, the mean of absolute SHAP value per feature was considered, such that

$$Mean\_Abs\_SHAP_{feature} = \frac{1}{S} \sum_{s=1}^{S} |SHAP_{feature}^s| \qquad (2)$$

where $SHAP_{feature}^s$ represents the SHAP value of a particular feature at sample s of the test set comprised of S samples. The absolute value is necessary to avoid canceling out during the averaging step, since the impacts can be positive or negative for each sample. To calculate the SHAP values, the SHAP Python library (SHAP Python, https://shap.readthedocs.io/ (accessed on 4 of November 2024) (version: 0.46.0)) was used.

3.2. Feature Selection with SHAP

Following the initial experiment, SHAP was employed to exclude features that exhibited a low contribution for the final forecast. For this purpose, a threshold of *k* features relative to the total features was considered, with the top *k* most important features carrying over to the next model iteration. The SHAP values were always calculated by the end of an iteration, allowing an iterative filtering and model assessment process. A performance threshold of *E*% against the baseline performance was used, considering the model trained with all the features. Should the performance decrease by a factor larger than stipulated, the process was halted and the previous iteration model was selected. Figure 2 illustrates the described methodology.
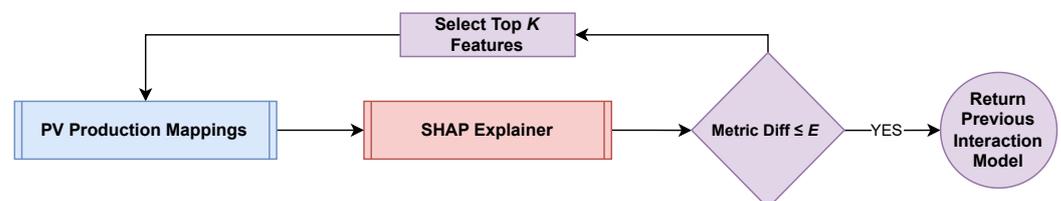
Figure 2. Proposed methodology for feature selection using SHAP values.

4. Evaluation Methodology

This section provides an overview of the algorithms and information regarding the dataset and data processing used throughout this work.

4.1. PV Production and Solar Irradiance Data

This paper used a real-world PV production dataset containing data for a three-phase installation in southern Madeira Island, Portugal. The PV production of this dataset was metered at 1 min intervals, spanning from January 2019 to March 2021 [38,39]. The installed

PV capacity was 3.92 kW and remained the same during this period. Solar irradiance and weather data were downloaded from SolCast [40] using the geographic coordinates of the PV installation to complement the PV data. The solar irradiance data were downloaded at a resolution of 30 min over the same period as the PV production data. This resolution was used because it matches SolCast's real-time API at the time of writing, allowing for future deployments that use this work's findings and are aligned with the needs of ongoing activities.

4.1.1. Data Pre-Processing

The resolution chosen for this work was 15 min, as this is a compromise between the 30 min resolution of the mentioned API and the resolution of the PV data in the dataset. Furthermore, a 15 min interval is adequate for day-ahead PV forecasting, because it balances the need for detailed, actionable information and the practical considerations of computational efficiency and grid operation requirements [21,41]. To achieve the desired temporal resolution, the PV data of the dataset were downsampled from their original resolution of 1 min by averaging the values over 15 min windows. For the data acquired from SolCast, the values were upsampled from the original resolution of 30 min to 15 min using forward filling.

Additional data cleansing was also necessary on top of the solar PV production data for the dataset. More precisely, a two-step approach was followed. First, identification of incorrect readings, e.g., PV production much higher than the PV nominal capacity (replaced with NaN) and PV production during the night (replaced with zero). Furthermore, during the night, the PV production data were consistently around -3.5 W, possibly due to the consumption of the PV system itself—these values were also set to zero.

4.1.2. Input Features

The dataset comprised all features made available by Solcast, complemented with four time-related features resulting from the timestamp's cosine and sine trigonometric transformations [42]. This information was added to the relay time information of the model cyclically, so that 23:30 is close to 00:00, and the 31st of December is numerically close to the 1st of January. For more information on the SolCast features, we refer to the documentation found on the website (SolCast documentation, https://solcast.com/irradiance-data-methodology(accessed on 4 of November 2024)).

These features are listed in Table 2. Figure 3a illustrates some domain features, and Figure 3b depicts the considered exogenous features. Day X and Day Y result from the transformation of a timestamp with regards to a day, while Year X and Year Y relay the cyclic nature of a year. All features used were normalized to a scale of 0 and 1.

Table 2. Domain and exogenous features used in this work.

Input Type	Features
Domain	AirTemp, AlbedoDaily, Azimuth, CloudOpacity, DewpointTemp, Dhi, Dni, Ebh, Ghi, GtiFixedTilt, GtiTracking, PrecipitableWater, RelativeHumidity, SnowDepth, SurfacePressure, WindDirection10m, WindSpeed10m, Zenith
Exogenous	Day X, Day Y, Day Z, Year X, Year Y

Experiments with tilted solar irradiation values were conducted to determine whether horizontal or tilted values should be used. To this end, models were trained using horizontal solar irradiance, tilted solar irradiance, and models with both tilted and horizontal solar irradiance. No statistical differences were found, and all features were preserved. Although

there was the possibility of feature redundancy, this presented itself as a scenario that could test whether the proposed methodology reduced model data requirements and lightened computational loads.

(**a**) Domain features   (**b**) Exogenous features
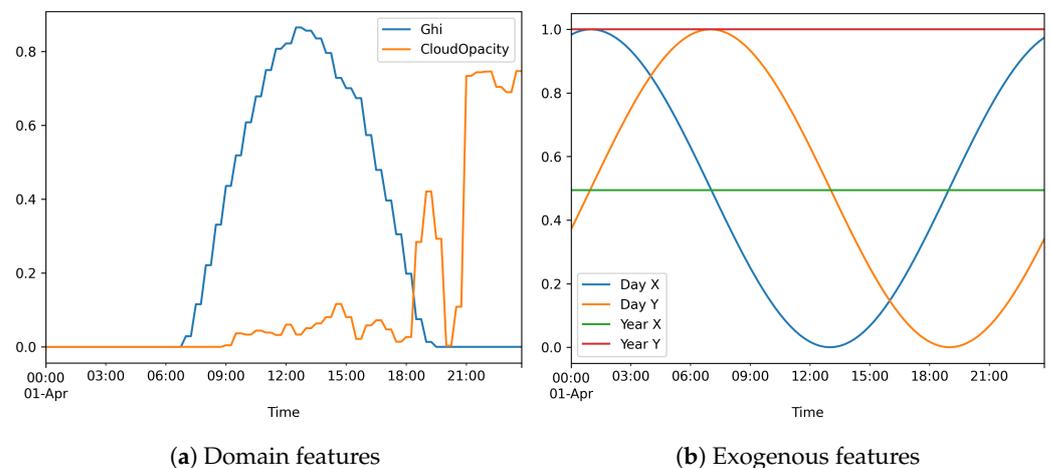
Figure 3. Examples of domain and exogenous features for a period of 24 h.

4.2. Experiments

This section presents the specifics of the individual experiments conducted to validate the proposed methods, following a top-down approach.

4.2.1. Year-Long and Seasonal Model Explanations

The application of the SHAP for model explanations was tested following a two-step approach. The first step relied on using the trained models and producing their predictions. These predictions were analyzed concerning their SHAP values, offering insights into the feature importance and impact on outputs. The goal of this experiment was to gain a global view of the impact distribution, as well as to observe prominent features. The second step consisted of analyzing the impact of seasonality on the feature importance. This approach offered more careful and in-depth observations of the SHAP results and established possible differences between seasons. This experiment aimed to understand if there were differences between the rankings of features across seasons, as well as their impact values. The test set was divided into seasons and analyzed separately to achieve this. Differences between seasons were reported, both using a single algorithm and between algorithms.

This experiment was conducted using a train–test split on the entire dataset. Specifically, the data for 2019 were used for training both algorithms, whereas the data for the first four months of 2020 were used for validation. The remainder of 2020 was utilized for testing.

4.2.2. Feature Selection and Benchmark

This experiment analyzed possible relationships between SHAP and performance, presenting a selection method to balance the model performance and computational load. Finally, a benchmark with two other feature selection methods was also presented to validate the proposed methodology. The selected benchmark feature selection methods were the *Spearman* correlation of features and output and the variance threshold method [43]. One important aspect is that these two methods were applied directly to the dataset and used earlier in the pipeline.

The first method relied on the *Spearman* correlation coefficient to determine whether a feature was related to the desired output. A threshold of 0.80 was established, with features presenting coefficients between −0.8 and 0.8 being discarded. The second method selected features based on how much variance a subset of features could explain. We set the threshold to 0.50, meaning that features incapable of explaining at least 50% of variance in the data were disregarded. Both the correlation and variance thresholds were set empirically. Afterward, new models were trained with the subset of selected features under the same conditions as the previous set. The findings were reported using the same metric to provide an accurate side-by-side representation of the effectiveness of the techniques.

A record of RAM usage was maintained to illustrate how feature selection could help reduce the computational load during model training. Although RAM may not be the ideal measure for assessing the full computational demands of model training, it served as the most directly comparable metric between the two models, given that different hardware resources were used for training and inference. Specifically, XGBoost was trained on the CPU, while TabNet was trained on the GPU.

5. Results and Discussion

5.1. Year-Long Effect of Input Features

The results obtained from Experiment 1 can be seen in Table 3. The features are organized by mean absolute SHAP values for both algorithms, and the original values are visible in Figure 4a,b. Each feature can be interpreted as contributing positively or negatively to model prediction depending on its SHAP value (horizontal axis), while the coloring indicates the real values of the features themselves. For example, when the Ghi feature has a strong positive value (magenta in the Figures) it influenced the output towards a higher prediction value. The reverse also holds, as when features present a low value (or even zero) they had a negative SHAP impact value and lower model forecast.

(**a**) XGBoost	(**b**) TabNet
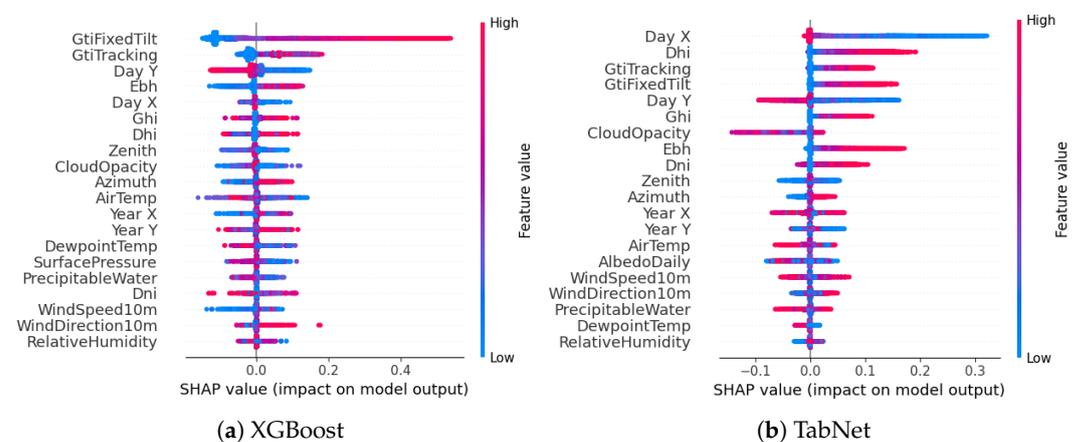
Figure 4. XGBoost and TabNet overall SHAP impact values. Each point represents an individual training example, with its color indicating the magnitude of a specific feature's value. The horizontal position of each point reflects the impact of that feature on the model's output.

The results of our experiment suggest that forms of irradiance (`Gti` and other related features) were indicated as being the most important to the trained models. The cosine transformation of the intra-day timestamp (`Day X`) was indicated as being the most relevant for TabNet, potentially acting as a substitute for the information of irradiation. Upon further inspection (also visible in Figure 3b), `Day X` assumes lower values during the portion of the day when sunlight is available.

Differences in feature importance across models can be attributed to model characteristics, intrinsic behaviors, and SHAP calculations. From a model perspective, TabNet

is designed to extract the maximum amount of information from a feature and combine it with others, to achieve good generalization through layers of abstraction characteristic of neural networks. From a SHAP calculation viewpoint, SHAP attributes impact values according to the direct impact on model outputs and feature influence on others. It is hypothesized that `Day X` may have been a feature that helps stabilize output values, acting as a regulator for other feature subsets. However, XGBoost did not consider `Day X` as important as TabNet, possibly due to the previously mentioned similarity of carried information relative to irradiation. Overall, the most important subset of features was a combination of irradiation values, time of day, and to some extent, cloud opacity.

Table 3. Rankings for feature impact by SHAP values.

	XGBoost		TabNet	
Rank	**Feature**	**Value**	**Feature**	**Value**
1	GtiFixedTilt	0.1316	Day X	0.0641
2	GtiTracking	0.0356	Dhi	0.0206
3	Day Y	0.0234	GtiTracking	0.0181
4	Ebh	0.0143	GtiFixedTilt	0.0173
5	Day X	0.0065	Day Y	0.0167
6	Ghi	0.0061	Ghi	0.0145
7	Dhi	0.0057	CloudOpacity	0.0118
8	Zenith	0.0048	Ebh	0.0113
9	CloudOpacity	0.0044	Dni	0.0057
10	Azimuth	0.0038	Zenith	0.0052
11	AirTemp	0.0036	Azimuth	0.0051
12	Year X	0.0035	Year X	0.0045
13	Year Y	0.0030	Year Y	0.0040
14	DewpointTemp	0.0027	AirTemp	0.0036
15	SurfacePressure	0.0026	AlbedoDaily	0.0036
16	PrecipitableWater	0.0023	WindSpeed10m	0.0028
17	Dni	0.0022	WindDirection10m	0.0027
18	WindSpeed10m	0.0021	PrecipitableWater	0.0023
19	WindDirection10m	0.0016	DewpointTemp	0.0021
20	RelativeHumidity	0.0013	RelativeHumidity	0.0015
21	AlbedoDaily	0.0005	SurfacePressure	0.0007
22	SnowDepth	0.0000	SnowDepth	0.0000

5.2. Seasonal Effect of the Input Features

XGBoost did not exhibit a great change in the rankings of the features when considering a more granular analysis, with the top 10 features remaining mostly unchanged. The middle set of features were the most affected by season changes, with the most significant being the air temperature, gaining importance during winter, while losing relevance during the remaining seasons. The `Day X` feature exchanged rankings with `Ghi`, further reinforcing our hypothesis of carrying similar information. The collected information on the seasonal rankings of features using XGBoost can be found in Table 4. Considering TabNet, a greater number of changes through seasons can be observed, largely maintaining the top subset of features. The most notable alterations of rankings were the increase in impact of cloud opacity during autumn and the overall increase in relevance of irradiance-related features during summer. Our findings using TabNet can be found in Table 5. Based on these findings, we can conclude that while the impact of features may vary seasonally, it is not significant enough to warrant a change in the way they are treated. These results are consistent with the findings in the previous section.

Table 4. XGBoost seasonal impact ranking changes.

Feature	Overall	Spring	Summer	Autumn	Winter
GtiFixedTilt	1	1 (-)	1 (-)	1 (-)	1 (-)
GtiTracking	2	2 (-)	2 (-)	2 (-)	2 (-)
Day Y	3	3 (-)	3 (-)	3 (-)	3 (-)
Ebh	4	4 (-)	4 (-)	4 (-)	4 (-)
Day X	5	6 (↓)	6 (↓)	5 (-)	5 (-)
Ghi	6	5 (↑)	5 (↑)	7 (↓)	7 (↓)
Dhi	7	7 (-)	8 (↓)	6 (↑)	6 (↑)
Zenith	8	8 (-)	9 (↓)	8 (-)	9 (↓)
CloudOpacity	9	9 (-)	7 (↑)	9 (-)	10 (↓)
Azimuth	10	11 (↓)	10 (-)	10 (-)	11 (↓)
AirTemp	11	15 (↓)	12 (↓)	12 (↓)	8 (↑)
Year X	12	10 (↑)	11 (↑)	11 (↑)	14 (↓)
Year Y	13	12 (↑)	13 (-)	14 (↓)	13 (-)
DewpointTemp	14	13 (↑)	18 (↓)	19 (↓)	12 (↑)
SurfacePressure	15	14 (↑)	15 (-)	13 (↑)	18 (↓)
PrecipitableWater	16	16 (-)	16 (-)	15 (↑)	15 (↑)
Dni	17	17 (-)	14 (↑)	16 (↑)	16 (↑)
WindSpeed10m	18	18 (-)	17 (↑)	17 (↑)	17 (↑)
WindDirection10m	19	19 (-)	19 (-)	18 (↑)	19 (-)
RelativeHumidity	20	20 (-)	20 (-)	20 (-)	20 (-)
AlbedoDaily	21	21 (-)	21 (-)	21 (-)	21 (-)
nowDepth	22	22 (-)	22 (-)	22 (-)	22 (-)

Table 5. TabNet seasonal impact ranking changes.

Feature	Overall	Spring	Summer	Autumn	Winter
Day X	1	1 (-)	1 (-)	1 (-)	1 (-)
Dhi	2	2 (-)	3 (↓)	2 (-)	2 (-)
GtiTracking	3	3 (-)	2 (↑)	5 (↓)	5 (↓)
GtiFixedTilt	4	5 (↓)	6 (↓)	4 (-)	3 (↑)
Day Y	5	4 (↑)	4 (↑)	6 (↓)	4 (↑)
Ghi	6	6 (-)	5 (↑)	7 (↓)	6 (-)
CloudOpacity	7	7 (-)	8 (↓)	3 (↑)	7 (-)
Ebh	8	8 (-)	7 (↑)	9 (↓)	8 (-)
Dni	9	10 (↓)	9 (-)	14 (↓)	12 (↓)
Zenith	10	9 (↑)	10 (-)	13 (↓)	11 (↓)
Azimuth	11	12 (↓)	15 (↓)	8 (↑)	9 (↑)
Year X	12	13 (↓)	14 (↓)	10 (↑)	10 (↑)
Year Y	13	11 (↑)	12 (↑)	11 (↑)	15 (↓)
AirTemp	14	16 (↓)	11 (↑)	16 (↓)	14 (-)
AlbedoDaily	15	17 (↓)	13 (↑)	12 (↑)	13 (↑)
WindSpeed10m	16	15 (↑)	19 (↓)	15 (↑)	16 (-)
WindDirection10m	17	14 (↑)	17 (-)	18 (↓)	17 (-)
PrecipitableWater	18	18 (-)	18 (-)	17 (↑)	18 (-)
DewpointTemp	19	19 (-)	16 (↑)	19 (-)	19 (-)
RelativeHumidity	20	20 (-)	20 (-)	20 (-)	20 (-)
SurfacePressure	21	21 (-)	21 (-)	21 (-)	21 (-)
SnowDepth	22	22 (-)	22 (-)	22 (-)	22 (-)

5.3. Feature Selection

The proposed methodology was applied once for each algorithm, with an established threshold of retaining the top 50% most impactful features. The following subset, sorted by impact, had the following iteration on XGBoost: GtiFixedTilt, GtiTracking, Day Y, Ebh,

`Ghi`, `AirTemp`, `Dhi`, `Zenith`, `CloudOpacity`, `Azimuth` and `Day X`. Considering TabNet, the sorted subset was composed as follows: `Ghi`, `Day X`, `GtiFixedTilt`, `Day Y`, `CloudOpacity`, `Zenith`, `GtiTracking`, `Dni`, `Azimuth`, `Ebh`, and `Dhi`.

TabNet presented the most changes against the baseline model impact distributions, with `Ghi` becoming the highest impact feature, followed by `Day X`, remaining unchanged throughout the seasons. XGBoost reported a similar impact distribution when compared to the baseline; however, `Day X` became the least impactful feature. This further solidifies our hypothesis of XGBoost interpreting `Day X` as a feature carrying information similar to irradiance.

Table 6 reports the findings relative to the application of the proposed methodology. These findings indicate that both models benefited from the presented methodology, exhibiting lower RMSE values. We attribute these improvements to the removal of noise, detrimental to model performance and leading to false relationships between features. Furthermore, the average usage of RAM while training is included to provide insights on the effects of feature removal in terms of computational resources, found in Table 7. Although it is not possible to draw straightforward conclusions regarding the RAM usage of the models, the overall tendency was to require fewer resources. We draw attention to the methodology including a hyperparameter optimization step that may have skewed results and produced larger models on subsequent iterations.

Table 6. RMSE benchmark using the proposed methodology.

Library	Overall	Spring	Summer	Autumn	Winter
XGBoost (baseline)	430.4	466.1	420.1	410.4	423.8
XGBoost (it. 1)	**413.4**	451.4	417.1	407.6	375.0
XGBoost (it. 2)	433.3	475.7	413.9	424.4	417.7
TabNet (baseline)	390.7	421.1	395.3	369.6	375.2
TabNet (it. 1)	**361.8**	391.9	315.8	373.4	363.4
TabNet (it. 2)	368.7	389.6	324.4	381.2	377.5

By conducting a second iteration of the proposed methodology, the overall RMSE values rose, although maintaining better performance against the baseline models. The sorted subset for the second iteration, considering XGBoost: `GtiFixedTilt`, `Day Y`, `GtiTracking`, `Ghi`, and `Ebh`, with the impact distribution being prevalent across seasons. TabNet considered the following sorted subset: `Ghi`, `Day X`, `GtiFixedTilt`, `Day Y`, and `CloudOpacity`, with the only alteration in ranking being the exchange of `Day X` and `GtiFixedTilt` during winter.

5.4. Benchmark

The variance feature selection method used for benchmarking presented a decrease in performance in all cases and poor overall results across seasons, with XGBoost models being the most affected, exhibiting higher RMSE values.

Feature selection through correlation presented a significant performance gap, with models with features selected through correlation performing worse than their baselines. The results for this experiment are presented in Table 8. Ultimately, these results can be attributed to the dataset being composed of a single PV installation, making it more susceptible to feature changes. However, while higher volatility increases the forecasting challenge, it further reinforces the importance of effective feature selection methods.

Table 7. Training RAM usage.

Library	Average RAM (MB)
XGBoost (baseline)	55.4
XGBoost (it. 1)	44.9
XGBoost (it. 2)	12.5
TabNet (baseline)	25.5
TabNet (it. 1)	45.0
TabNet (it. 2)	17.8

Table 8. Feature selection methods benchmark RMSE results.

Library	Overall	Spring	Summer	Autumn	Winter
XGBoost (baseline)	**430.4**	466.1	420.1	410.4	423.8
XGBoost (corr)	509.6	528.4	523.3	497.0	490.0
XGBoost (var)	1023.8	1168.6	931.2	890.5	1107.9
TabNet (baseline)	**390.7**	421.1	395.3	369.6	375.2
TabNet (corr)	398.6	410.2	394.7	392.9	397.0
TabNet (var)	988.8	1068.5	1140.8	843.8	890.0

For comparative purposes, a day of the testing set is provided in Figure 5. The day 4th of August was selected, as it presents a smooth PV generation profile that allows the exemplification of the effectiveness of the proposed methodology. It can be observed that TabNet closely followed the ground-truth data, as well as the first iteration of the methodology. XGBoost and its respective iteration followed the general trend but exhibited higher errors across the entire production cycle. The second iteration of TabNet showcased the need for stopping criteria, presenting performance degradation.
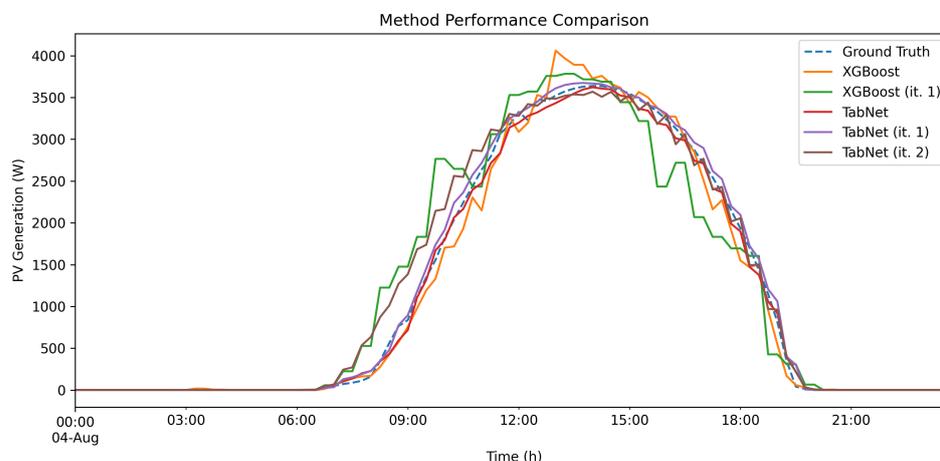
Figure 5. Model performances on a summer day for the testing set (4 August 2020).

While several studies have explored forecasting models, direct comparisons with other works are challenging, due to differences in used data and metrics. Notably, studies delving into PV and XAI have typically reported R^2 values, while this work presents RMSE. Despite these differences, the proposed methodology remains applicable and capable of being integrated into other forecasting frameworks.

6. Conclusions

This work proposed a methodological framework for applying XAI to PV forecasting algorithms using SHAP local explanations. The framework was applied to a real-world dataset to understand how the feature important would vary throughout the year. The

obtained results showed that irradiance features like `Gti` were the most important for the models, with TabNet also highlighting the cosine transformation of the intra-day timestamp (`Day X`) as a substitute for irradiance. XGBoost's top features remained stable across seasons, though air temperature gained relevance in winter, while TabNet showed more seasonal changes, such as increased cloud opacity in autumn.

The SHAP values were further leveraged to help in feature selection, where the 50% most important feature subset was kept and used to train subsequent models. The results obtained show that for XGBoost, the key features included `GtiFixedTilt`, `GtiTracking`, and `Day Y`, while TabNet's subset prioritized `Ghi`, `Day X`, and `GtiFixedTilt`. Overall, TabNet showed the most changes, with `Ghi` as the top feature, whereas XGBoost treated `Day X` as least important, reinforcing the idea that it carried similar information to irradiance.

One limitation of this work is that the data were collected on Madeira Island, where proximity to the Equator may have reduced the seasonality variability. Future studies should apply the methodology in other locations to validate its effectiveness and the seasonality conclusions drawn here. Another limitation is the narrow focus on resource usage, as only RAM consumption during model training was analyzed. Future studies should collect a broader range of resource data, to better understand computational costs.

Further research is needed to explore the methodology's potential in areas beyond PV mapping and forecasting, such as net-load forecasting. A sensitivity analysis of the threshold parameters would be beneficial, allowing for fine-tuning, rather than relying on the 50% threshold used in this study. Additionally, applying other forecasting strategies, like probabilistic methods, could enhance understanding of the methodology's impact.

Abbreviations

The following abbreviations are used in this manuscript:

LIME	Local Interpretable Model-Agnostic Explanation
ML	Machine Learning
PV	Solar Photovoltaic
RMSE	Root Mean Squared Error
SHAP	SHapley Additive exPlanations
XAI	eXplainable Artificial Intelligence

References

1. Gandhi, O.; Kumar, D.S.; Rodríguez-Gallegos, C.D.; Srinivasan, D. Review of Power System Impacts at High PV Penetration Part I: Factors Limiting PV Penetration. *Sol. Energy* **2020**, *210*, 181–201. https://doi.org/10.1016/j.solener.2020.06.097.
2. Pierro, M.; De Felice, M.; Maggioni, E.; Moser, D.; Perotto, A.; Spada, F.; Cornaro, C. Photovoltaic Generation Forecast for Power Transmission Scheduling: A Real Case Study. *Sol. Energy* **2018**, *174*, 976–990. https://doi.org/10.1016/j.solener.2018.09.054.
3. Machado, E.; Pinto, T.; Guedes, V.; Morais, H. Electrical Load Demand Forecasting Using Feed-Forward Neural Networks. *Energies* **2021**, *14*, 7644. https://doi.org/10.3390/en14227644.
4. Rozas, W.; Pastor-Vargas, R.; García-Vico, A.M.; Carpio, J. Consumption–Production Profile Categorization in Energy Communities. *Energies* **2023**, *16*, 6996. https://doi.org/10.3390/en16196996.
5. Chaudhary, P.; Rizwan, M. Energy management supporting high penetration of solar photovoltaic generation for smart grid using solar forecasts and pumped hydro storage system. *Renew. Energy* **2018**, *118*, 928–946. https://doi.org/10.1016/j.renene.2017.10.113.
6. Sheik Mohammed, S; Titus, F.; Thanikanti, S.B.; Sulaiman, S.M.; Deb, S.; Kumar, N.M. Charge Scheduling Optimization of Plug-In Electric Vehicle in a PV Powered Grid-Connected Charging Station Based on Day-Ahead Solar Energy Forecasting in Australia. *Sustainability* **2022**, *14*, 3498. https://doi.org/10.3390/su14063498.
7. Gunning, D.; Aha, D. DARPA's Explainable Artificial Intelligence (XAI) Program. *AI Mag.* **2019**, *40*, 44–58. https://doi.org/10.1609/aimag.v40i2.2850.
8. Machlev, R.; Heistrene, L.; Perl, M.; Levy, K.Y.; Belikov, J.; Mannor, S.; Levron, Y. Explainable Artificial Intelligence (XAI) techniques for energy and power systems: Review, challenges and opportunities. *Energy AI* **2022**, *9*, 100169. https://doi.org/10.1016/j.egyai.2022.100169.
9. Mitrentsis, G.; Lens, H. An interpretable probabilistic model for short-term solar power forecasting using natural gradient boosting. *Appl. Energy* **2022**, *309*, 118473. https://doi.org/10.1016/j.apenergy.2021.118473.
10. Nallakaruppan, M.K.; Shankar, N.; Bhuvanagiri, P.B.; Padmanaban, S.; Bhatia Khan, S. Advancing solar energy integration: Unveiling XAI insights for enhanced power system management and sustainable future. *Ain Shams Eng. J.* **2024**, *15*, 102740. https://doi.org/10.1016/j.asej.2024.102740.
11. Rizk-Allah, R.M.; Abouelmagd, L.M.; Darwish, A.; Snasel, V.; Hassanien, A.E. Explainable AI and optimized solar power generation forecasting model based on environmental conditions. *PLoS ONE* **2024**, *19*, e0308002. https://doi.org/10.1371/journal.pone.0308002.
12. Petrosian, O.; Zhang, Y. Solar Power Generation Forecasting in Smart Cities and Explanation Based on Explainable AI. *Smart Cities* **2024**, *7*, 3388–3411. https://doi.org/10.3390/smartcities7060132.
13. Utama, C.; Meske, C.; Schneider, J.; Schlatmann, R.; Ulbrich, C. Explainable artificial intelligence for photovoltaic fault detection: A comparison of instruments. *Sol. Energy* **2023**, *249*, 139–151. https://doi.org/10.1016/j.solener.2022.11.018.
14. Kuzlu, M.; Cali, U.; Sharma, V.; Guler, O. Gaining Insight Into Solar Photovoltaic Power Generation Forecasting Utilizing Explainable Artificial Intelligence Tools. *IEEE Access* **2020**, *8*, 187814–187823. https://doi.org/10.1109/ACCESS.2020.3031477.
15. Sarp, S.; Kuzlu, M.; Cali, U.; Elma, O.; Guler, O. An Interpretable Solar Photovoltaic Power Generation Forecasting Approach Using An Explainable Artificial Intelligence Tool. In Proceedings of the 2021 IEEE Power Energy Society Innovative Smart Grid Technologies Conference (ISGT), Washington, DC, USA, 16–18 February 2021. https://doi.org/10.1109/ISGT49243.2021.9372263.
16. Sobri, S.; Koohi-Kamali, S.; Rahim, N.A. Solar photovoltaic generation forecasting methods: A review. *Energy Convers. Manag.* **2018**, *156*, 459–497. https://doi.org/10.1016/j.enconman.2017.11.019.
17. Al-Shahri, O.A.; Ismail, F.B.; Hannan, M.A.; Lipu, M.S.H.; Al-Shetwi, A.Q.; Begum, R.A.; Al-Muhsen, N.F.O.; Soujeri, E. Solar photovoltaic energy optimization methods, challenges and issues: A comprehensive review. *J. Clean. Prod.* **2021**, *284*, 125465. https://doi.org/10.1016/j.jclepro.2020.125465.
18. Dimitropoulos, N.; Sofias, N.; Kapsalis, P.; Mylona, Z.; Marinakis, V.; Primo, N.; Doukas, H. Forecasting of short-term PV production in energy communities through Machine Learning and Deep Learning algorithms. In Proceedings of the 2021 12th International Conference on Information, Intelligence, Systems Applications (IISA), Chania Crete, Greece, 12–14 July 2021; pp. 1–6. https://doi.org/10.1109/IISA52424.2021.9555544.
19. Li, M.; Wang, W.; He, Y.; Wang, Q. Deep learning model for short-term photovoltaic power forecasting based on variational mode decomposition and similar day clustering. *Comput. Electr. Eng.* **2024**, *115*, 109116. https://doi.org/10.1016/j.compeleceng.2024.109116.
20. Das, U.K.; Tey, K.S.; Seyedmahmoudian, M.; Mekhilef, S.; Idris, M.Y.I.; Van Deventer, W.; Horan, B.; Stojcevski, A. Forecasting of photovoltaic power generation and model optimization: A review. *Renew. Sustain. Energy Rev.* **2018**, *81*, 912–928. https://doi.org/10.1016/j.rser.2017.08.017.
21. Reindl, T.; Walsh, W.; Yanqin, Z.; Bieri, M. Energy meteorology for accurate forecasting of PV power output on different time horizons. *Energy Procedia* **2017**, *130*, 130–138. https://doi.org/10.1016/j.egypro.2017.09.415.

22. Gomes, L.; Morais, H.; Gonçalves, C.; Gomes, E.; Pereira, L.; Vale, Z. Impact of Forecasting Models Errors in a Peer-to-Peer Energy Sharing Market. *Energies* **2022**, *15*, 3543. https://doi.org/10.3390/en15103543.

23. Bae, K.Y.; Jang, H.S.; Jung, B.C.; Sung, D.K. Effect of Prediction Error of Machine Learning Schemes on Photovoltaic Power Trading Based on Energy Storage Systems. *Energies* **2019**, *12*, 1249. https://doi.org/10.3390/en12071249.

24. De Hoog, J.; Perera, M.; Ilfrich, P.; Halgamuge, S. Characteristic profile: Improved solar power forecasting using seasonality models. *ACM Sigenergy Energy Inform. Rev.* **2021**, *1*, 95–106. https://doi.org/10.1145/3508467.3508476.

25. Huang, C.J.; Kuo, P.H. Multiple-Input Deep Convolutional Neural Network Model for Short-Term Photovoltaic Power Forecasting. *IEEE Access* **2019**, *7*, 74822–74834. https://doi.org/10.1109/ACCESS.2019.2921238.

26. Limouni, T.; Yaagoubi, R.; Bouziane, K.; Guissi, K.; Baali, E.H. Accurate one step and multistep forecasting of very short-term PV power using LSTM-TCN model. *Renew. Energy* **2023**, *205*, 1010–1024. https://doi.org/10.1016/j.renene.2023.01.118.

27. Barredo Arrieta, A.; Díaz-Rodríguez, N.; Del Ser, J.; Bennetot, A.; Tabik, S.; Barbado, A.; Garcia, S.; Gil-Lopez, S.; Molina, D.; Benjamins, R.; et al. Explainable Artificial Intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI. *Inf. Fusion* **2020**, *58*, 82–115. https://doi.org/10.1016/j.inffus.2019.12.012.

28. Hosain, M.T.; Jim, J.R.; Mridha, M.; Kabir, M.M. Explainable AI approaches in deep learning: Advancements, applications and challenges. *Comput. Electr. Eng.* **2024**, *117*, 109246. https://doi.org/10.1016/j.compeleceng.2024.109246.

29. Rožanec, J.; Trajkova, E.; Kenda, K.; Fortuna, B.; Mladenić, D. Explaining Bad Forecasts in Global Time Series Models. *Appl. Sci.* **2021**, *11*, 9243. https://doi.org/10.3390/app11199243.

30. Jaakkola, T.; Alvarez Melis, D. Towards robust interpretability with self-explaining neural networks. *Neural Inf. Process. Syst. (NIPS)* **2018**, *31*.

31. Selvaraju, R.R.; Cogswell, M.; Das, A.; Vedantam, R.; Parikh, D.; Batra, D. Grad-CAM: Visual Explanations from Deep Networks via Gradient-based Localization. *Int. J. Comput. Vis.* **2020**, *128*, 336–359. https://doi.org/10.1007/s11263-019-01228-7.

32. Ribeiro, M.T.; Singh, S.; Guestrin, C. "Why Should I Trust You?": Explaining the Predictions of Any Classifier. *arXiv* **2016**, arXiv:1602.04938. https://doi.org/10.48550/arXiv.1602.04938.

33. Lundberg, S.M.; Lee, S.I. A Unified Approach to Interpreting Model Predictions. In Proceedings of the Advances in Neural Information Processing Systems, Long Beach, CA, USA, 4–9 December 2017; Curran Associates, Inc.: New York, NY, USA, 2017; Volume 30.

34. Chen, T.; Guestrin, C. XGBoost: A Scalable Tree Boosting System. In Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '16, San Francisco, CA, USA, 13–17August 2016; Association for Computing Machinery: New York, NY, USA, 2016; pp. 785–794. https://doi.org/10.1145/2939672.2939785.

35. Arik, S.O.; Pfister, T. TabNet: Attentive Interpretable Tabular Learning. *arXiv* **2020**, arXiv:1908.07442. http://doi.org/10.48550/arXiv.1908.07442.

36. Akiba, T.; Sano, S.; Yanase, T.; Ohta, T.; Koyama, M. Optuna: A Next-generation Hyperparameter Optimization Framework. *arXiv* **2019**, arXiv:1907.10902. https://doi.org/10.48550/arXiv.1907.10902.

37. Abadi, M.; Barham, P.; Chen, J.; Chen, Z.; Davis, A.; Dean, J.; Devin, M.; Ghemawat, S.; Irving, G.; Isard, M.; et al. TensorFlow: A System for Large-Scale Machine Learning. In Proceedings of the 12th USENIX Symposium on Operating Systems Design and Implementation (OSDI 16), Savannah, GA, USA, 2–4 November 2016; pp. 265–283.

38. Pereira, L.; Cavaleiro, J.; Morais, H. Understanding the Role of Solar PV and Battery Energy Storage in a Snack Bar: A Case Study in Madeira Island. In Proceedings of the 2023 IEEE 21st International Conference on Industrial Informatics (INDIN), Lemgo, Germany, 18–20 July 2023; pp. 1–7. https://doi.org/10.1109/INDIN51400.2023.10218295.

39. PRSMA; M-ITI; EEM; ACIF-CCIM. *Installation Report of the DSM Demo (Final Version)*; Technical Report 4.11; European Commission: Funchal, Portugal, 2021.

40. Bright, J.M. Solcast: Validation of a satellite-derived solar irradiance dataset. *Sol. Energy* **2019**, *189*, 435–449. https://doi.org/10.1016/j.solener.2019.07.086.

41. Pedro, A.; Krutnik, M.; Yadack, V.M.; Pereira, L.; Morais, H. Opportunities and Challenges for Small-Scale Flexibility in European Electricity Markets. *Util. Policy* **2023**, *80*, 101477. https://doi.org/10.1016/j.jup.2022.101477.

42. Faustine, A.; Pereira, L. FPSeq2Q: Fully Parameterized Sequence to Quantile Regression for Net-Load Forecasting with Uncertainty Estimates. *IEEE Trans. Smart Grid* **2022**, *13*, 2440–2451. https://doi.org/10.1109/TSG.2022.3148699.

43. Chandrashekar, G.; Sahin, F. A survey on feature selection methods. *Comput. Electr. Eng.* **2014**, *40*, 16–28. https://doi.org/10.1016/j.compeleceng.2013.11.024.