



Local-global methods for generalised solar irradiance forecasting

Timothy R. Cargan¹ · Dario Landa-Silva¹ · Isaac Triguero^{1,2,3}

Accepted: 3 January 2024
© The Author(s) 2024

Abstract

For efficient operation, solar power operators often require generation forecasts for multiple sites with varying data availability. Many proposed methods for forecasting solar irradiance / solar power production formulate the problem as a time-series, using current observations to generate forecasts. This necessitates a real-time data stream and enough historical observations at every location for these methods to be deployed. In this paper, we propose the use of Global methods to train generalised models. Using data from 20 locations distributed throughout the UK, we show that it is possible to learn models without access to data for all locations, enabling them to generate forecasts for unseen locations. We show a single Global model trained on multiple locations can produce more consistent and accurate results across locations. Furthermore, by leveraging weather observations and measurements from other locations we show it is possible to create models capable of accurately forecasting irradiance at locations without any real-time data. We apply our approaches to both classical and state-of-the-art Machine Learning methods, including a Transformer architecture. We compare models using satellite imagery or point observations (temperature, pressure, etc.) as weather data. These methods could facilitate planning and optimisation for both newly deployed solar farms and domestic installations from the moment they come online.

Keywords Deep learning · Time series forecast · Solar irradiance forecast · Generalised model · Local-global

1 Introduction

Solar power is an increasingly used source of renewable electricity. In the case of photovoltaic (PV) solar, irradiance (power per unit area radiated from the sun [1]) is converted into electricity. Power output typically tracks the sun, peaking in the middle of the day. However, changes in

weather/atmospheric conditions can cause output to fluctuate abruptly throughout the day [2]. However, energy generating companies must know their power output in advance of its generation. Any unplanned deviation can push the grid out of equilibrium with the grid operator passing balancing costs onto offending producers [3]. The variability of PV solar makes accurate and timely forecasts vital for the efficient operation of grid connected solar [4, Chapter 11.2]. It is common for generators to have facilities in multiple locations. This requires a forecast of solar irradiance for every area of interest (AOI). It is likely for each AOI to have different levels of data availability, for both historic and real time values. This inconsistency can present challenges in developing forecasting models for each AOI.

One challenge is that historical and real time irradiance data might not be available for each AOI. Whilst new solar installations can be fitted with equipment capable of relaying irradiance observations, retrofitting existing plants can be cost-prohibitive, with domestic installations presenting an even larger challenge to be practical in “production”. Many solar irradiance forecasting methods in the literature make use of time-series methods and so rely on a continuous stream of irradiance data to be available at the AOI [5].

✉ Timothy R. Cargan
timothy.cargan@nottingham.ac.uk
Dario Landa-Silva
dario.landasilva@nottingham.ac.uk
Isaac Triguero
isaac.triguero@nottingham.ac.uk

¹ Computational Optimisation and Learning (COL) Lab,
School of Computer Science, University of Nottingham,
Nottingham, NG8 1BB, United Kingdom

² Department of Computer Science and Artificial Intelligence,
University of Granada,, Granada,, Spain

³ DaSCI Andalusian Institute in Data Science
and Computational Intelligence, University of Granada,,
Granada, Spain

Another challenge is that as solar installations become more widespread, solar irradiance forecasting is required for a larger number of AOIs. Developing and maintaining a “Local” model for each AOI is not practical; for deploying forecasting on a domestic level this could mean potentially thousands of models. Additionally, each AOI requires enough historical data to effectively learn a model. Many works in the literature [6, 7] focus on providing forecasts for a single location. A generalised “Global” model considers multiple AOIs but still relies on historic and real time data being available for all of them [6, 7].

This paper makes a contribution by addressing the two above challenges and proposes forecasting methods less reliant on real time irradiance data and suitable for practical application to multiple AOIs. The methods leverage different modalities of weather data and “Global” models to forecast solar irradiance even when some of the AOIs may not have real time irradiance data, without sacrificing overall performance.

Incorporating weather data can help generate more reliable irradiance forecasts. Access to point-based weather data (temperature, wind speed etc) is available from commercial providers for practically any location [8, 9]. This approach has been used successfully by [10, 11]. Additionally, real-time satellite imagery from providers such as [12] or whole sky images provide an alternative way to potentially capture richer weather information, as demonstrated in [13, 14].

While an individual AOI may lack data, as the number of AOIs increases, so does the aggregate data quality and availability. This paper leverages this fact by using “Global” models, creating a single model to predict for multiple AOIs instead of facing the challenges of managing multiple Local models. Global models have several advantages. They can result in higher quality forecasts by learning from multiple locations’ data [15, 16]; Data collected at defunct AOIs be incorporated into the model further to improve performance; Learning from multiple time series reduces dependencies on any individual AOI, allowing forecasts to be uncoupled from historical or local telemetry data, which can be unreliable or non-existent.

This paper presents methods for solar irradiance forecasting when working with multiple AOIs. It proposes an extension to the Global model, reducing its reliance on real-time streams of irradiance, and facilitating the inclusion of locations with limited data. To evaluate the method, a novel encoder-decoder transformer architecture for solar irradiance forecasting from satellite imagery is used in addition to typical machine learning (ML) methods. A detailed statistical analysis is performed comparing the proposed method against Local and Global models. In addition, different modalities of weather data are compared; and the importance of both real-time irradiance, and weather data is assessed.

The methodology presented in this paper compares Local and Global forecasting methods applied to several ML techniques to predict solar irradiance. Moreover, the Global approach is extended to circumvent some of the real-world data dependency issues and analyse the effect of data availability on performance. These extensions are Global Plant Holdout (GPH) to handle a lack of historic data and Global Plant kNear (GPkN) to cope with missing real-time data.

The proposed methodology can provide practical industry applicable approaches useful for short to medium-range (intra-day) planning. The proposed methods are validated using irradiance measurements and corresponding weather data from 20 AOIs distributed across the UK (see Fig. 2). Performance of the models is compared across four training schemes, Local, Global, GPH and GPkN, outlined in Section 3.2. More specifically, the paper:

- Shows that Global models can leverage data from multiple locations for improved forecasting performance and can generalise to unseen locations removing the need for historical data to predict at unseen AOI.
- Proposes a method to uncouple irradiance predictions from real-time observations by substituting observations from nearby plants.
- Presents an encoder-decoder Transformer and compares it to a number of standard ML methods commonly used for forecasting Irradiance (DNN, CNN, LSTM and Trees).
- Compares the use of satellite images to point weather data and shows that using rich weather data from satellites can produce better forecasts.

The remainder of this paper is structured as follows; Section 2 provides an overview of the literature. Section 3 describes our proposed methods while Section 4 outlines our experimental framework as well as gives details of the data used. In Section 5 we present our results and analyse the performance of our proposed methods. Finally, Section 6 concludes the paper.

2 Background

In this section, we provide background information on the problem domain and give an overview of the various ML based methods used for irradiance forecasting. Section 2.1 defines variables used throughout the paper and formalises the irradiance forecasting problem. Section 2.2 reviews the various ML based irradiance forecasting methods, discussing their advantages and their main caveats. Section 2.3 discusses how existing global models handle AOIs with limited data.

2.1 Problem definition and notation

The variables we use in equations and diagrams throughout our paper are as follows:

- t – Denotes a time step; t_0 refers to *now* from the models perspective with t_1, t_2, \dots, t_n being in the future (forecasts) and $t_{-1}, t_{-2}, \dots, t_{-n}$ in the past (observations).
- I – A sequence of irradiance observations, made of elements i .
- \hat{I} – A sequence of irradiance predictions, made of elements \hat{i} .
- W – A sequence of weather data, this can be point weather or satellite imagery; elements w are themselves a set of multiple observations or pixel values (see Section 4.1).
- C – A sequence of calculated values; elements c are composed of metadata deterministically calculated based on an AOI location and the time (see Table 4).
- X – Used to denote an arbitrary variable.
- f – Used to denote a function.
- m – Used to denote a parameterised model that approximates f .
- ww – Warm-up window, the number of preceding time steps from t_0 that the model sees.
- fh – Forecast horizon, the number of time steps from t_0 that the model predicts.

Using the above notation, we can define our goal in its simplest form as follows: At time step t , predict future irradiance values $I_t = [i_{t+1}, \dots, i_{t+fh}]$. If we assume there is a function that can map an input X_t to irradiance forecasts $f(X_t) = I_t$, we can define an ML problem to approximate the function f using a model m such that $m(X_t) \rightarrow \hat{I}_t$. Here X represents any data that could be used by a model such as weather data, historic irradiance observations, or calculated values.

2.2 Machine learning methods

There are multiple ML based methods of irradiance forecasting in the literature. These can be loosely divided into two types: regression-based (Section 2.2.1) and time-series based (Section 2.2.2). However, many newer methods combine elements from both approaches (Section 2.2.3).

2.2.1 Regression

Regression models use correlated values to make predictions. As previously noted, the predominant variable¹ in how much solar irradiance makes it to the ground is the weather [4, Chapter 11.2]. Since weather and irradiance are correlated,

¹ for the same time of day, day of year and location

a regressive model can be learned to predict irradiance using weather data. We formulate the regression problem as; given a set of weather values, e.g. $w_t = [wind, temp, pres]$; we aim to create a model where $m(w_t) \rightarrow \hat{i}_t$, mapping weather state to irradiance predictions.

There are many examples throughout the literature of irradiance forecasts being created from regressing over weather data using a variety of techniques [5, 10, 11, 17, 18]. Although classical ML approaches such as Support Vector Regression and Decision Trees are actively used for irradiance forecasting [19, 20]. In recent years, neural networks (NNs) have proved to be a highly effective tool for regression due to their power and flexibility as function estimators [21]. Accordingly, a number of NN based methods have been employed [11, 17] with convolutional neural networks (CNNs) being used to regress over satellite images to produce predictions [13].

An advantage of modelling irradiance predictions as a regression problem using weather data is that predictions are uncoupled from real-time irradiance observations such as in [6]. However, regressive models typically map input features at step t to an output prediction at the same step. In order to forecast irradiance, future weather states are needed. While it is possible to offset inputs and outputs e.g. $m(x_t) \rightarrow \hat{i}_{t+1}$ [22], many regressive methods rely on the use of externally provided weather forecasts [19, 23].

A limitation on the use of weather forecasts is that updated predictions can only be made when new weather forecasts are provided. As such, the forecast used can limit the timeliness of a model's predictions. For example, if a model produces an hourly forecast with a 12-hour horizon using weather data that updates once every 6 hours, while predictions are made for all steps, there are times when the forecast is stale and its useful horizon reduced. For example, if a prediction was made at 6 am, covering all steps until 6 pm, by 9 am there are only 9, dated, forecast steps remaining. As such, weather data sources used must be carefully considered as they can affect both the performance and capability of models.

The use of weather forecasts can introduce another level of uncertainty to the models.

2.2.2 Time series

Time-series modelling is an alternative method for building forecasting models. A time series is a sequence of data where the order of observations matters, typically there exists a sequential relationship between examples [24]. Time series forecasting methods use previous observations of the value the model intends to predict as input. In the case of our irradiance forecasting problem, observations from the last few hours would be used to predict the future values in the sequence. Using the notation from above we can formalise the ML problem as: given a sequence of irradiance measures,

$I_t = [i_{-ww}, \dots, i_0]$; We aim to create a model m such that $m(I_t) \rightarrow [\hat{i}_1, \dots, \hat{i}_{fh}]$.

There are numerous methods used throughout the literature for time series forecasting on a wide array of problems [25–29]. We split these further into two approaches, autoregressive and sequence modelling.

Autoregressive approaches use a lagged window over a number of the previously seen examples to create a forecast [30]. There are several examples of purely autoregressive models being used for irradiance forecasting. These include classical statistical methods such as ARIMA [31], as well as neural networks [22]. The length of the window is a hyperparameter that must be tuned to provide models appropriate context, however, the computational complexity increases in line with window length. While effective, autoregressive methods can only model sequences that fall within the lagged window. As such, they can fail to capture relationships that are beyond the length of the window [24].

Sequence modelling can solve the issue of reliance on the lagged window, by modelling an evolving state. With the rise of deep learning, recurrent neural networks (RNNs) have become a popular way to do so. In order to create forecasts, RNNs process elements sequentially. At each step, t , taking both input features X_t , and the models' previous state α_{t-1} from the preceding step as input, and outputs a prediction, \hat{I}_t and creating new state α_t .

By passing in its previous state information can be transmitted from one time-step to another. Long short-term memory (LSTM) architectures have proved to be a highly effective form of RNN, by selectively taking in its state from previous time steps they are able to model sequence dependencies [24, 32]. Some examples of LSTMs being used to generate irradiance forecasts can be found in [33, 34].

Despite their effectiveness, both autoregressive and sequence modelling approaches require access to real-time data to make predictions. From a practical standpoint, this limits the applicability of purely time series models to locations that can provide a feed of real-time data. However, it is also possible to create models that leverage both kinds of data. Many of the more recent methods we have classed as time series are in fact hybrid, using both irradiance and weather data [28, 34].

2.2.3 Transformer

The transformer architecture is a highly flexible neural network architecture. Through the use of an order invariant attention mechanism combined with various positional encodings, the transformer architecture is capable of being applied to both regression and time-series problem formulations. The transformer block takes as input a sequence

of tokens (vectors) and outputs a sequence of the same length. In the case of an encoder-decoder transformer, the encoder transforms the input sequence into an internal representation. The decoder, using cross-attention to extract information from the output of the encoder, autoregressively creates a new output sequence token by token. Originally introduced as a method for sequence-to-sequence machine translation [35], the transformer has also proved successful at regression tasks such as image classification with the vision transformer (ViT) [36]. For encoding an image, ViT splits it into patches that are linearly projected to vectors that are used as input tokens. This method of patching and linear projection of images used by ViT has become a standard method for encoding images for use with transformers.

The few examples of Transformers being used for irradiance forecasting in the literature have given mixed results. Using a sequence of irradiance data [37, 38] made use of an aggressive transformer for hourly irradiance forecasting. However, [22] found the Transformers performance to be worse than other typical ML methods using a similar aggressive model for high-frequency short-range forecasting (30 seconds). Both [39] and [14] found the use of ViT style transformers trained on all-sky-imagery for irradiance forecasting to be successful showing improvements on benchmark data sets.

2.3 Global models and missing data

Using Local models, developing irradiance forecasting models for AOIs with limited or missing data is a challenge. While works such as [40] enable predictions for locations with limited history by selectively extracting extra data from correlated locations, they are still Local models.

When working with multiple AOIs, it is possible to learn a single Global model fit on all AOIs [15, 23] By training a single global model, the hope is that it will generalise well and be able to forecast at an unseen location. In [16, 41], this is referred to as cross-learning.

This idea was first explored in [6], by leveraging a combination of satellite observations and irradiance forecasts from the European center for medium-range weather forecasts (ECMWF) a Global DNN model was trained. However, the ECMWF forecasts update every 6 hours hence limiting the ability of their DNN model to output timely forecasts. Here we propose making use of standard, and widely commercially available, weather forecasts. Many of these updates sub-hourly removing any data dependency [8].

[41] use weather forecasts to create a number of Global models using a number of ML methods to generate predictions at locations with no historic data. Whilst their approach allows for predictions in locations without real-time irradiance observations, they achieve this simply by not including them. We explore this as well as propose an alternative

method, GPkN, to circumvent the data dependency by substituting values.

3 Motivation and methodology

In this section, we discuss our proposed methodology, outlining how we address the irradiance forecasting problem defined in Section 2.1. In Section 3.1 we state our motivation while, Section 3.2 describes our proposed data models to solve the issues and limitations set out above.

3.1 Motivation

Our goal is to present plausible methods, capable of being deployed in an industry setting that provide timely, accurate forecasts of irradiance when dealing with multiple AOIs. To that end, our model must: (1) Produce forecasts at a frequency and with a forecast horizon to be of practical use; (2) Generalise across locations; (3) Uncouple irradiance forecasts from real-time observations. When designing our methods we must consider the various possible technical limitations when selecting input features, i.e. availability, update frequency, timeliness, etc.

Given these objectives and limitations, we use models capable of predicting with an hourly frequency with a forecast horizon of 6 hours. We show that predictions of steps after 6 hours are predominately dominated by the weather data and as such in “production” will be limited by weather forecast accuracy.

3.2 Data models / training schemes

The data available at both training and inference times dictate the overall design of any forecasting model. Broadly, we consider two classes of data when designing the models:

1. Historic observations - this is data that is guaranteed to be available at training time. This would be a database of weather and irradiance values for one or more AOIs spanning multiple years.
2. Real-time observations - the data used to make the forecasts. It consists of telemetry/observations transmitted in a timely manner (on the order of minutes) to the model in order for it to produce useful forecasts. This could be an observation of irradiance measured at a given AOI or weather forecasts sourced from 3rd parties.

Both classes of data are needed to create a useful forecasting model. However, for any given AOI there may be limitations on data availability. Using the problem definition in Section 2.1, we outline four data models to train a forecasting model using various combinations of possibly available

Table 1 Three AOIs are given, x , y and z , with y the closest AOI to x

Data Model	Train Input		Inference Input	
	Irradiance	Weather	Irradiance	Weather
Local	I_x	W_x	I_x	W_x
Global	I_x, I_y, I_z	W_x, W_y, W_z	I_x	W_x
GPH	I_y, I_z	W_y, W_z	I_x	W_x
GPkN	I_y, I_z	W_y, W_z	I_y	W_x

Overview of the inputs used by the four schemes to train a model to predict for AOI x

data. Each represents a conceivable dataset that could be available for a group of AOIs when attempting to build a forecasting model.

In this context, AOIs data consists only of irradiance values, it is assumed that weather data (point or satellite) will always be available, both historically and in real time, for any AOIs location.

The four data models are Local, Global, Global Plant Holdout (GPH) and Global Plant kNear (GPkN). Local is a typical approach with Global the logical way to generalise across locations solving some of the limitations of a Local model. GPH and GPkN are further extensions of the Global approach each solving a data limitation. Table 1 gives an overview of data used by the four methods at training and inference time.

3.2.1 Local

Local Models, as their name suggests, are localised to a single AOI, and is the approach common in the literature [17]. A local model is trained on data from and produces predictions for a single AOI. Since local models are created by fitting a model using only data from a given AOI, one model must be created per AOI. This makes the Local approach most practical when there are a small number of AOIs. However, the lack of data dependency between models means computation can be scaled out during both the training and inference phases. While being a relatively straightforward approach, Local models have three key limitations:

1. They require enough historic data for each AOI to effectively learn a model (the amount of historic data needed is dependent on the model and data being used).
2. A model per AOI must be generated. If there are a large number of AOIs, such as in the case of a domestic solar fleet, computation and storage constraints could become an issue.
3. A real-time data feed of irradiance for all AOIs is needed.

3.2.2 Global

The Global model is a generalised version of the local model, a single model that can predict for all AOIs [6]. It is created by training a single instance of the model on the union of data for every AOI. Global models solve two of the limitations listed above faced by local models.

1. Since they are trained on multiple AOI it is possible to create an effective model even if some AOIs have limited historical data.
2. A single model artefact is created, greatly reducing the overhead of managing multiple models,

However, the third limitation above remains because the Global model still assumes the *best case* scenario where there is access to ample historic data and a real-time feed of irradiance for all AOIs. Because of this, the Global approach is, by definition, limited to AOIs with both historic and real-time data. Additionally, introducing a data dependency between all AOIs causes overall model training time to increase. In practice, depending on the distribution and number of AOIs a down-sampling approach could be used to reduce training time. For example, reducing the number of training samples used per AOI based on the density of AOIs in its geographic area. Unlike the Local approach, due to data dependencies, it is challenging to use the scale-out approach when training a single model [42]. However, the computation can still be scaled out during inference time.

3.2.3 Global plant holdout (GPH)

Lack of historic data is a common occurrence, it would be the case for a new installation or an older domestic AOI without telemetry recordings. The GPH approach eliminates the need for all AOIs to have historic data. Like the Global approach, a single model is trained using the data from all AOIs with available historic data (even if only partial). The model is then used to make predictions for all AOIs both with and without historic data, using their real-time data.

This solves the need for all AOIs to have historic data. Another advantage of GPH is that data from decommissioned AOIs can still be used to train the models. However, for this approach to be viable, any models created must be able to generalise well to the unseen AOIs.

To simulate this, a standard cross-validation approach is taken. Each AOI is randomly assigned to one of 5 folds. At train time, for each fold a model is created using data from all but the AOIs in the given fold. At test time only the AOIs in the fold are evaluated. This process is repeated for each fold resulting in a worst-case prediction for every AOI in the training set.

3.2.4 Global plant kNear (GPkN)

GPkN attempts to solve the *worst case* scenario where we have neither access to historic irradiance measures nor real-time data for the AOI. While an unlikely scenario, it could occur in the case of a sensor failure. It is also conceivable in the domestic setting, or if attempting to estimate the output of an AOI. As it represents the most challenging scenario, it sets a baseline of model performance. Like GPH, a single model is trained using data from the AOIs with historic data. To generate predictions for the AOIs with no data, we substitute the real-time values from the nearest AOI with data.

To evaluate GPkN, the same per-fold GPH model was used, however, the real-time irradiance values are substituted to the nearest plant not in the hold-out fold. Distances were calculated using the haversine function.

4 Experimental framework

In this section, we present our experimental framework. In Section 4.1 we provide the details of the raw data used as well as any preprocessing that was done. In Section 4.2 we describe the models used and their configurations. Finally, Section 4.3 explains the error metrics and validation methods used.

4.1 Data details

We use irradiance data, point-based weather observations, and satellite imagery. The raw data is sourced from three unique providers (see Table 2) with each data set covering the period 2015-01-01T00:00 to 2021-01-01T00:00. All data is aligned to the hour and updates with a one-hour frequency. Any time steps with missing data, from any source, are omitted.

Irradiance data We sourced irradiance values from the “MIDAS Open: UK hourly solar radiation data” [43]. It consists of hourly irradiance ($\frac{\text{kilojoules}}{\text{square meter}}$) from over 80 locations distributed throughout the UK. Each location consists of a time series with data for all or part of the period. We selected a subset of 20 locations to use as our primary AOIs, the locations were selected as they have the fewest missing values

Table 2 Overview of the datasets used

Data	Source	Freq	Modality	Resolution
Irradiance	[43]	1h	Point	-
Weather Data	[8]	1h	Point	-
Satellite Data	[12]	1h	Image	3km

All datasets cover the time period 2015-01-01T00:00 to 2021-01-01T00:00

for the timespan. This was done to as fair as possible comparisons when evaluating local models between AOIs.

Pre-processing – The raw irradiance values are normalised using the function:

$$norm(i) = \begin{cases} \ln(1 + i) - 4, & \text{if } i \geq 20 \\ -3, & \text{otherwise} \end{cases} \quad (1)$$

This was done to give an approximately normal distribution centred on 0 for daytime hours, while raw values of less than 20 are clipped to the lowest value as we consider them night-time.

Point Weather data Point weather data comprised of observations of variables such as wind speed, temperature, etc. made at an observation stations In the UK there exists a large fleet of stations distributed all over the country. Historic, hourly, and point weather observations for all locations within the irradiance dataset were sourced from a commercial supplier [8].

Pre-processing – The features were normalised using either the Z-score, defined as $zscore(x) = \frac{x-\mu}{\sigma}$ where μ is the mean value of the feature and σ is its standard deviation, or log normalisation. A full list of features used and their normalisation is given in Table 3.

Satellite Data Historic hourly satellite imagery is sourced from EUMETSAT [12]. Satellite data consists of observations of several wavelengths of light and is presented as an image. The raw data for each time step consists of a set of 12 images each 500px X 500px covering the UK area [-12.0W, 48.0S, 5.0E, 61.0N], with each pixel being approximately 3km by 3km. The area processed is shown in Fig. 1 with the location of the AOIs highlighted. The raw images cover 11 channels spanning wavelengths from visible to infrared and an additional 12th channel in the visible spectrum at a higher resolution. An example for each wavelength at day and night is shown in Fig. 2.

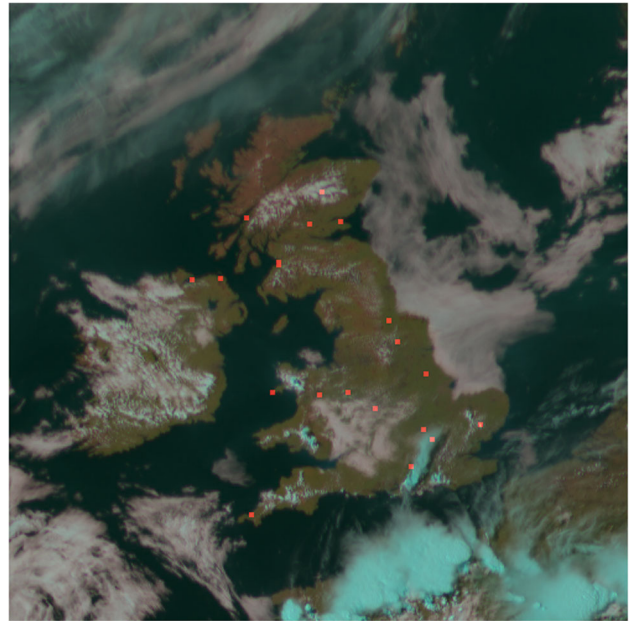


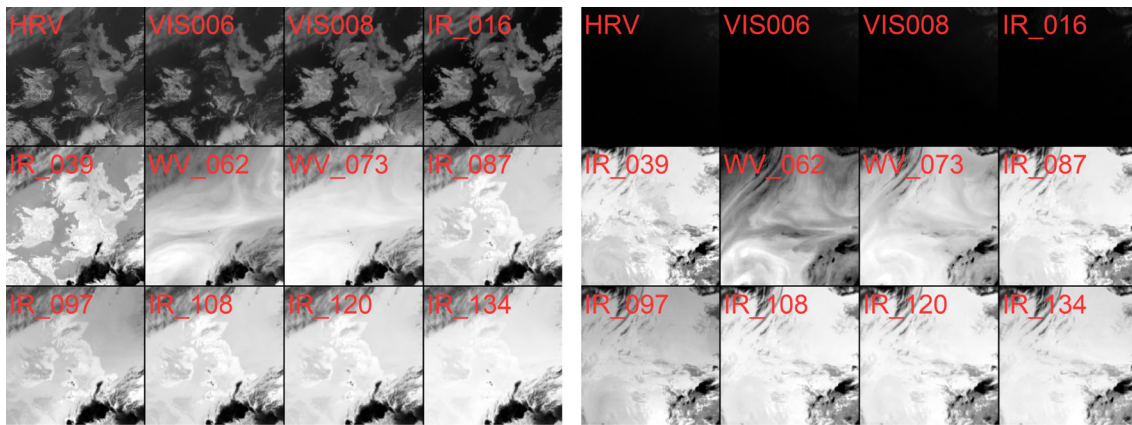
Fig. 1 UK area of visible wavelengths processed to RGB with the locations of the AOIs highlighted in red

Pre-processing – For each AOI a 16px X 16px image centred on the given AOI is cropped from the full image. The raw pixel values are normalised to the range 0-1.

Calculated Values In addition to the three data sources above, a fourth pseudo data source of values calculated from the AOIs latitude, longitude and datetime are used. The generated features are outlined in Table 4. These values help give the models context for when and where the forecast is being generated. As these values can be calculated simply from the AOI location and time it is assumed they are always available.

Table 3 The point weather features and respective normalisation methods used

Feature	Unit	Normalisation
Cloud Cover	%	Z-score Mean: 60, std: 30
Clear-sky Irradiance (GHI)	$\frac{W}{m^2}$	ln
Precipitation	$\frac{mm}{hr}$	Z-score Mean: 0.1, std: 0.33
Pressure	mb	Z-score Mean: 1000, std: 15.5
Relative humidity	%	Z-score Mean: 82, std: 13
Temperature	°k	Z-score Mean: 283, std: 5.5
N/S component of wind	$\frac{m}{s}$	-
E/W component of wind	$\frac{m}{s}$	-



(a) Day time satellite images (2018-05-28T15:00)

(b) Night-time satellite images (2018-05-28T03:00)

Fig. 2 A sample of each wavelength of the satellite images covering the full UK area

4.2 Model configurations

In order to evaluate the different data models outlined in Section 3.2, and the various possible input features described in Section 4.1, we utilise five ML methods: CNNs, LSTMs, DNNs, Transformers, and Trees. Each learning model aims to approximate a function that maps its inputs to irradiance values. The DNNs, Transformers, and Trees are purely regressive methods generating forecasts in a single step. The CNN and LSTM contain elements of both regressive and time series approaches with explicit architectures to capture the sequential nature of the data and generate their forecasts autoregressively, using previous outputs as inputs to generate the next step. All methods make use of both the real-time irradiance and the calculated values as input features. However, the modality of the weather data used is model dependant. Each model's weather data modality is shown in Table 5.

An effort was made to ensure the selected hyperparameters and architectural decisions will produce results indicative of the approach's possible performance, however, no full

hyperparameter search was performed. Better values and architectures may exist. The same model configuration was used regardless of the data model or input features where possible. The architecture and hyperparameters selected for each model are outlined below.

4.2.1 Trees

Trees, or tree-based models, are a classical ML approach. We use a random forest (RF), an ensemble model, training multiple decision trees each on subsets of the training data and combining the predictions from each tree to produce the final output [45]. For the rest of this article, we use the term Trees interchangeably with Random Forests. Trees were chosen as they have been used effectively used on numerous forecasting problems. Their use of an ensemble approach makes them highly robust to over-fitting. They are also easy to implement with many standard library implementations.

The trees were trained using Python 3.9 and TensorFlow Decision Forests 1.1.0 [46]. The following hyper-parameters were used for all experiments in Table 6

Table 4 Calculated features, all solar positions were calculated using pvlib [44]

Feature	Description
Year sin	$\sin(ts * 2\pi / \text{Year Seconds})$
Year cos	$\cos(ts * 2\pi / \text{Year Seconds})$
Day sin	$\sin(ts * 2\pi / \text{Day Seconds})$
Day cos	$\cos(ts * 2\pi / \text{Day Seconds})$
Solar altitude	$\sin(\text{SolarElevationAngle})$
Azimuth sin	E/W of the position of the sun in the sky
Azimuth cos	E/W of the position of the sun in the sky

ts is the unix timestamp in seconds

Table 5 The modality of weather data used by each of the learning models

Model	Weather Modality
CNN	Image
DNN	Point
LSTM	Point
Transformer	Image
Tree	Point

Table 6 Hyperparameters used by the random forests

Hyper Parameter	Value
Number of Trees	20
Min Examples per Leaf	2
Max Depth	32

Forecasting It is possible for a tree-based model to output multiple labels [47, 48]. A limitation of the implementation we used for our tree models is that they can only output a single value. This means that unlike the other deep learning (DL) approaches used, in order to produce forecasts a unique model was trained for each step in the forecast horizon $t_1 \dots t_{fh}$.

4.2.2 Deep neural network (DNN)

A Fully connected DNN, also referred to as a multi-layer perception (MLP), consists of an input layer, followed by a number of fully connected hidden layers, and an output layer. Each layer is made of several units, each of which outputs a weighted sum of all the previous layers' outputs. Between each layer is a non-linear activation function such as a ReLU. During the training phase, the weights are adjusted using backpropagation in order to minimise a loss function, such as mean square error (MSE), with respect to training data.

In our case, the DNN uses the point weather data, past irradiance and the calculated features concatenated together into a single input vector. The final output layer is forecast horizon units wide producing all fh outputs at the same time. A diagram of the architecture is shown in Fig. 3.

4.2.3 Long short-term memory (LSTM)

Using the same data as the DNN and Tree models, the LSTM autoregressively generates irradiance predictions. For the initial input steps (of length ww), the LSTM is in a warm-up phase, using the past observations of both weather and irradiance data to establish its internal state. Once the past data has been consumed, the model enters the prediction phase, where for each prediction step, t_i , the previous output of the model at t_{i-1} is used in place of observed irradiance.

The architecture of our LSTM network is shown in Figure 4. At each step, t , the weather features are passed through an MLP before being concatenated with the calculated features and irradiance value. This is followed by the LSTM layer and then another MLP. The final layer has 1 unit to produce the irradiance prediction for step $t + 1$.

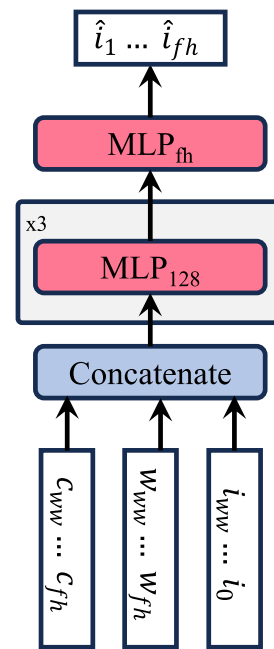


Fig. 3 The DNN architecture used in this paper. Each of the 3 hidden layers is 128 units wide and uses ReLU activation. The final MLP block has no activation function

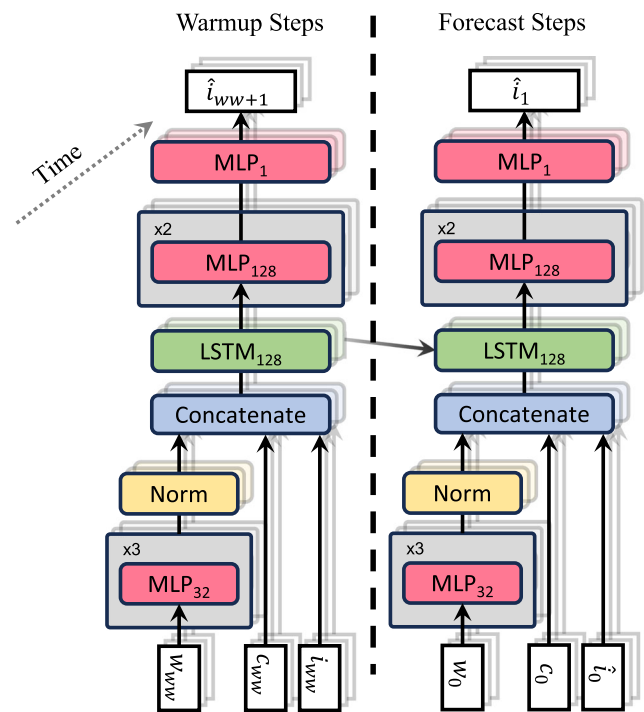


Fig. 4 Overview of the LSTM autoregressive architecture. Note that all weights are the same for every time step. The weather MLP has 3 layers of 32 units. The LSTM layer has 128 units. The second MLP block has 2 hidden layers each 128 units

4.2.4 Convolutional neural network (CNN)

Unlike the other DL based models, the CNN use satellite images rather than point weather data. However, they still incorporate both the real-time irradiance and calculated values as inputs. The images for each time step are stacked into a 4D tensor of shape [timesteps, height, width, channels], resulting in a standard weather state input with a shape: $(fh + ww, 16, 16, 12)$.

The CNN model generates forecasts using a convolutional head to extract information from the raw satellite images into an internal representation vector. The vector is then used in place of the point weather data, concatenated with irradiance and calculated features, and passed into a model with the same architecture as the LSTM.

The architecture of the convolutional head is shown in Fig. 5. The images are initially grouped by channels and passed into a stack of three convolutions with a kernel size of $(3 \times 3 \times 3)$ convolving over both space and time with 16 features and a ReLU activation. The outputs of each group are recombined and passed into three blocks of; convolutions $(3 \times 3 \times 3)$ with 16 features, activation (ReLU), and max pool $(1, 2, 2)$ reducing just spatial dimensions. The outputs are then flattened and passed to an LSTM module.

4.2.5 Transformer

Our transformer-based architecture is based on the Vision Transformer (ViT) [36]. The raw satellite images are first processed using the same convolutional head as the CNN (Fig. 5), the output of which is split into patches using the same method as that of the ViT. The patches are flattened, projected, and summed with learned positional encoding for use as input tokens for an encoder-decoder transformer to forecast irradiance. In addition to image tokens, the calculated values and historical irradiance for each time step are added as extra tokens to the encoder. We use a dimensionality of 256 for the transformer tokens with both the encoder and decoder having 4 heads and 4 layers. An overview of the architecture is shown in Fig. 6.

4.2.6 Training

All models were trained with a warmup window (ww) of 12 steps and a forecast horizon (fh) of 6 steps. All the DL models were trained for 20 epochs using a batch size of 128, the optimiser settings for each model are specified in Table 7. The models were trained using Python 3.10, Jax 4.19, Flax 0.7.4, Optax 0.1.17. Our code can be found at <https://github.com/timcargan/local-global-solar>.

Table 7 Optimiser setting used in the DL models

Model	Optimiser	Learning Rate
DNN	Adam	0.0003
LSTM	Adam	0.0003
CNN	Adam	0.0003
Transformer	AdamW	0.0009

Additionally, for the transformer cosine learning rate decay was used, with 200 warm-up steps and 10,000 decay steps; An initial learning rate of $1e-7$ and a final learning rate of $3e-5$ were used

4.3 Metrics and validation

There exist numerous metrics for evaluating the performance of regression models throughout the literature. They primarily provide a summary of the error distribution where the error is defined as the difference between the observed and predicted value [49, 50]. Additionally, there exist many ways to measure forecast accuracy [51]. A common feature of forecast errors is a scaling of the error enabling comparison across distributions.

Using our stated notation, we define our error metrics as follows. For a dataset comprised of n examples where $I = [i_1, \dots, i_n]$ is the true irradiance value and $\hat{I} = [\hat{i}_1, \dots, \hat{i}_n]$ the predicted values. The following performance metrics defined below are used to evaluate and compare the various models:

- Normalised Root Mean Square Error (nRMSE)

$$\text{nRMSE} = \frac{\text{RMSE}}{\bar{I}} \quad (2)$$

Where: \bar{I} is the mean of the irradiance sequence and RMSE is simply the Root Mean Square Error, $\sqrt{\frac{1}{n} \sum_{t=1}^n (i_t - \hat{i}_t)^2}$.

- Forecast Skill (S)

$$S = \frac{1}{p} \sum_{w=0}^p \left(1 - \frac{U_w}{V_w} \right) \quad (3)$$

Where: The dataset is split into p periods of l elements. U_w represents the error of the models' forecasts for a given period, calculated as the root mean square of the error scaled by the clear sky (GHI): $\sqrt{\frac{1}{l} \sum_{t=1}^l \left(\frac{i_t - \hat{i}_t}{\text{GHI}_t} \right)^2}$. V_w is a measure of the forecast difficulty for a given period calculated as the root mean square of the clear sky scaled variability of irradiance: $\sqrt{\frac{1}{l} \sum_{t=2}^l \left(\frac{i_{t-1}}{\text{GHI}_{t-1}} - \frac{i_t}{\text{GHI}_t} \right)^2}$. S is in turn calculated as the average of the ratio of error to difficulty for all the periods

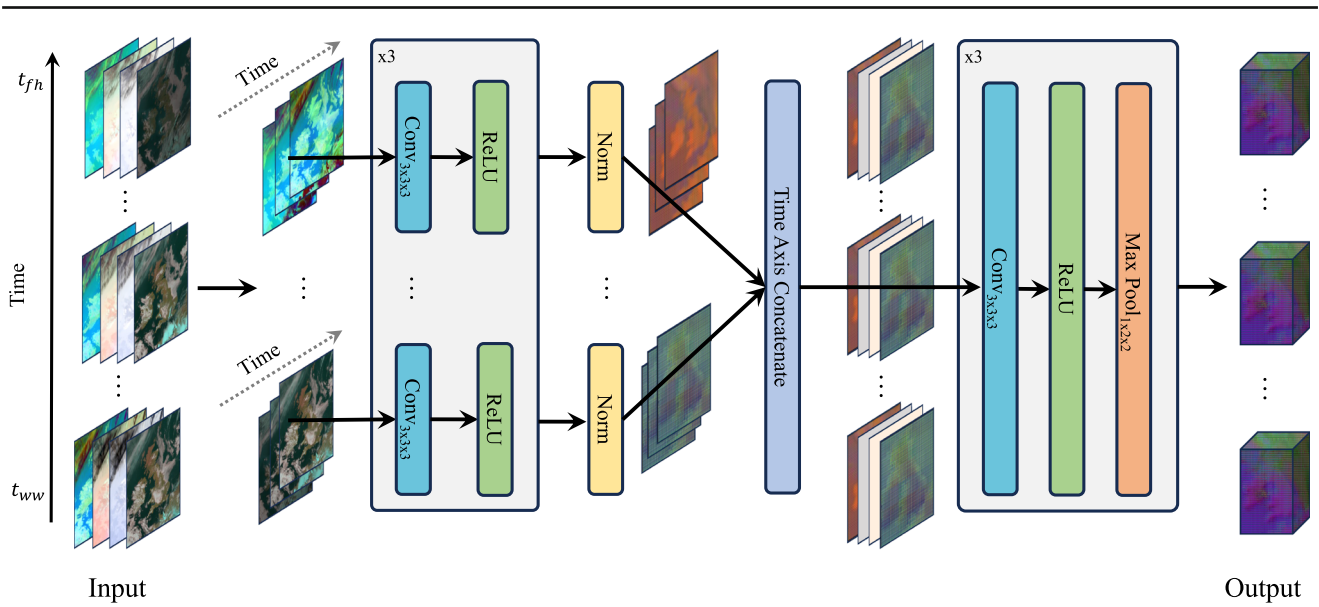


Fig. 5 The convolutional head architecture, used by both the CNN and Transformer models. The channels are groped as (HRV, VIS006, VIS008, IR_016), (WV_062, WV_073), (IR_097, IR_108, IR_120, IR_134) with IR_039 and IR_087

omitted. In the case of the CNN the output are flattened and fed into an LSTM in place of the weather inputs. For the Transformer the output is treated as an image

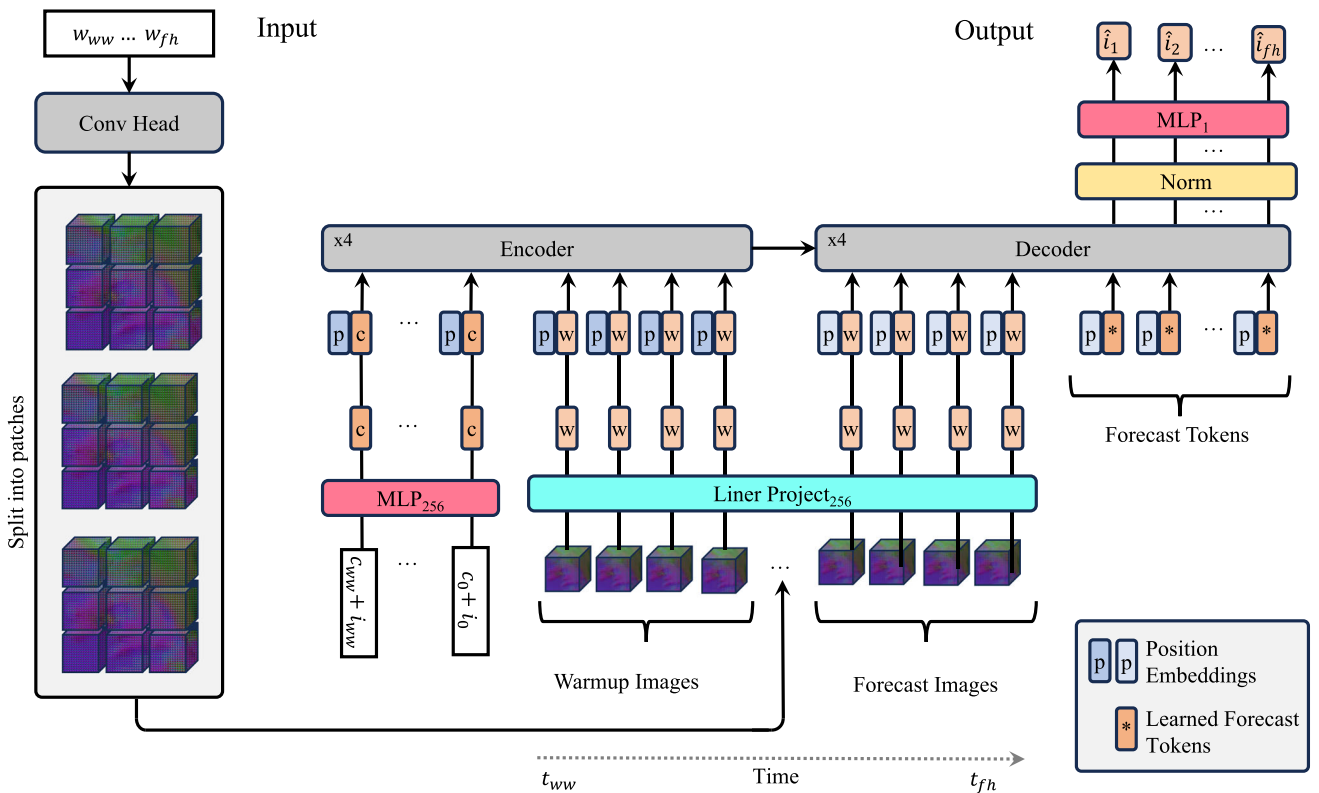


Fig. 6 An overview of the Transformer architecture. The output of the convolutional head is processed as images in a ViT, split into patches and linear projected into tokens. A learned positional encoding is added

to each token followed by a standard encoder-decoder transformer. For the decoder fh extra tokens are added and their output is used as the forecast

Table 8 A description of forecast skill interpretation

Value	Interpretation
1	The prediction is perfectly accurate.
0	The prediction is no better than that of a persistence model using the ratio of the last observed irradiance to clear sky $\hat{y} = \frac{I_{t-1}}{GHI_{t-1}} GHI_0$.
< 0	Negative values indicate that the prediction is worse than the persistence model.

within the dataset. We used a period size of 1 calendar month.

Both nRMSE and S are scale-invariant enabling a nuanced comparison between models and various output distributions. nRMSE was selected as it has been widely used in the literature. While being scale-invariant, absolute errors are punished the same regardless of the size of the target, i.e. a prediction of 15 for a true value of 10 results in the same error as a prediction of 105 for a true value of 100. S was proposed by [52] and is similar to mean absolute scaled error, adjusting in proportion to the size of the target sequence however the metric also factors in a measure of forecast difficulty.

A note on the error metrics and their interpretation S and nRMSE are interpreted in inverse of one another. In the case of nRMSE - a lower value is better with 0 indicating the predictions are perfectly accurate. A value of 1 would mean forecasts are very bad as the RMSE is equal to the mean of the sequence, as such the average absolute error is the same as the mean of the sequence. S, conversely, is interpreted with a higher value indicating better performance. S can fall in the range $(-\infty, 1]$ although a value less than 0 indicates poor performance. A more detailed interpretation is in Table 8.

nRMSE punishes errors equally regardless of the size of the target. As such, a good nRMSE indicates that the magnitude of the errors is consistently small i.e. $\hat{y} = y \pm 30$.

A low nRMSE error but poor forecast skill could indicate that the model performs poorly when the target irradiance values are low, early and late in the day e.g. the model always predicts 50 above the true value. It could be due to a low variance in targets vs GHI making it 'easy' to predict the sequence. Conversely, a higher nRMSE but good forecast skill could indicate that the models performed well during the day when target values are higher e.g. the model was always 10% over the target value, or that the sequence was challenging to predict, leading to larger errors.

All errors presented are calculated using only daytime values. We defined daytime as any point where the target irradiance, $y > 20$ and $GHI > 1$. We use both conditions to minimise the risk of any sensor errors sewing the results.

4.3.1 Validation

For Local and Global models, the data was split into train and test partitions of roughly 70% train and 30% test. The data was split on 1st May 2019 with all models being trained on data from before the split date and evaluated on data after. The date was arbitrarily chosen from a previously used dataset.

For both GPH and GPkN we use a standard 5-fold cross-validation approach to generate predictions for all 20 AOIs. The AOIs are randomly split into five folds. For each fold, we train a model on data from the 16 out-of-fold AOIs using data from before the cutoff date and test on the remaining in-fold 4 AOIs using data after the cutoff date. We note that the same AOI-fold assignment was used for all experiments

4.4 Statistical tests

We use two non-parametric statistical tests for hypothesis testing to give statistical support when analysing our results [53]. We use non-parametric tests as the initial conditions required for parametric tests to be reliable may not be met. For pairwise comparisons, we make use of the Wilcoxon test [54, 55]. We assume a level of significance of $\alpha = 0.1$ To evaluate our methods against one another we use the Friedman test [56] to identify statistical differences among them. We use the Holm post-hoc test to determine which algorithms have significant differences among the $1 * n$ comparisons [57].

A note on Weather Forecasts

In order to evaluate our models we use historical observation in place of actual weather forecasts. This was done to remove a degree of uncertainty caused by any error in the weather forecast as we attempt to understand the effect different input features can have. As such the results presented are the best case for any given method and in production using real weather forecasts we would expect a drop in performance depending on the accuracy of the forecasts.

5 Analysis of results

In this section we present our results and analysis. Section 5.1 addresses the effect different input features have on performance. Using a single learning model, we analyse performance for both the Local and Global flavours. In Section 5.2, we compare the five learning models in both Local and Global versions; We also analyse the importance of AOI location and look at how the use of different modalities of weather data affects performance. Finally, in Section 5.3 we show that GPH and GPkN flavours can be used to circum-

Table 9 Average of nRMSE (lower is better) and S (higher is better) across all AOIs and forecast steps showing the effect of using various input features with a DNN. We boldface the overall best Local and Global results for both the nRMSE and S

	nRMSE				S			
	Local		Global		Local		Global	
	mean	std	mean	std	mean	std	mean	std
All	0.389	0.081	0.376	0.064	0.708	0.067	0.745	0.047
Irrad	0.512	0.095	0.479	0.098	0.666	0.078	0.691	0.078
Calc	0.632	0.073	0.631	0.038	0.590	0.098	0.606	0.081
Weather	0.417	0.098	0.376	0.067	0.711	0.053	0.717	0.049

vent data limitations and further analyse the importance of real-time irradiance.

The raw results and our source code are available at <https://github.com/timcargan/local-global-solar>. In addition, we include supplementary material with further experiments.

5.1 Effect of inputs

In this section, we analyse the effect inclusion of different input features has on learning model performance. We focus on the DNN due to their easy ability to change input features and speed to train. As outlined in Section 4.1, there are various potential input features that could be used by the models. We group them into three classes: (1) Calculated values such as location, time, solar position etc. (2) Real-time observations of irradiance values. (3) Weather features, past observations as well as forecasts of future states. We trained the DNN using four combinations of these inputs:

- All – Irradiance, weather and calculated values
- Irradiance – Irradiance and calculated values

- Calc – Only the calculated values
- Weather – Weather and calculated values

These four input combinations were selected to cover the range of possible data availability; from a full data set with both historic and real-time values for every AOI to no data for any given AOI.

In Table 9 we present results for all AOIs averaged over the full forecast horizon (forecast step 1 to 6). We further break down the results by per forecast step in Fig. 7 and AOI in Fig. 8. Additionally, Table 10 shows the results of the Friedman test ranking the input combinations at forecast steps one and six.

Use of just the calculated data produced the worse results by a significant margin, consistently ranking last for both flavours and time steps. Conversely, using all the inputs produced the best overall results in almost all cases. All ranked best or had no significant difference from the best-ranked method for every case tested.

Using either weather or irradiance as additional data combined with the calculated features resulted in performance

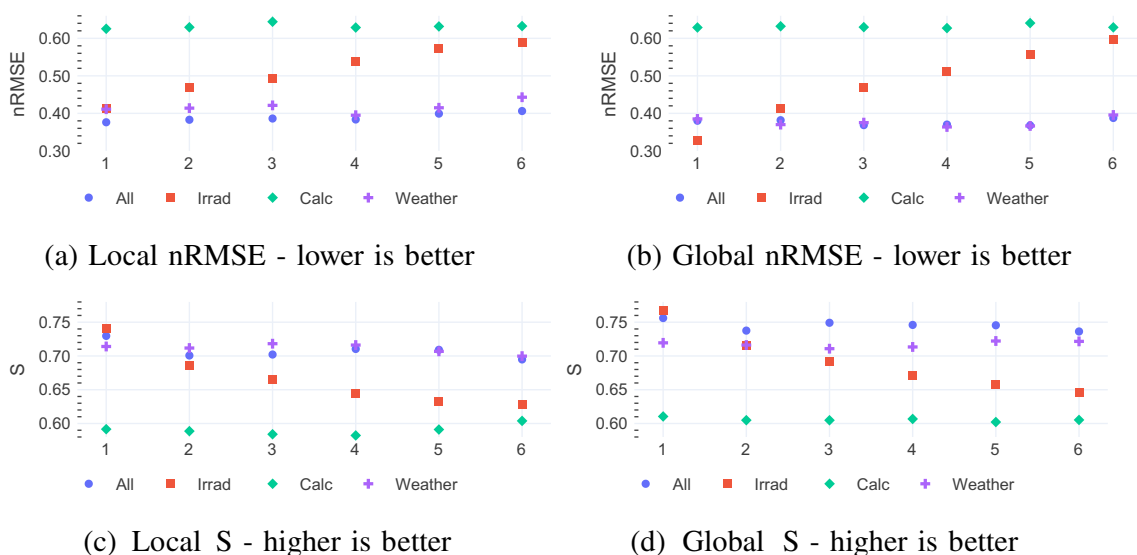


Fig. 7 Distribution of error per step for each of the inputs feature groups

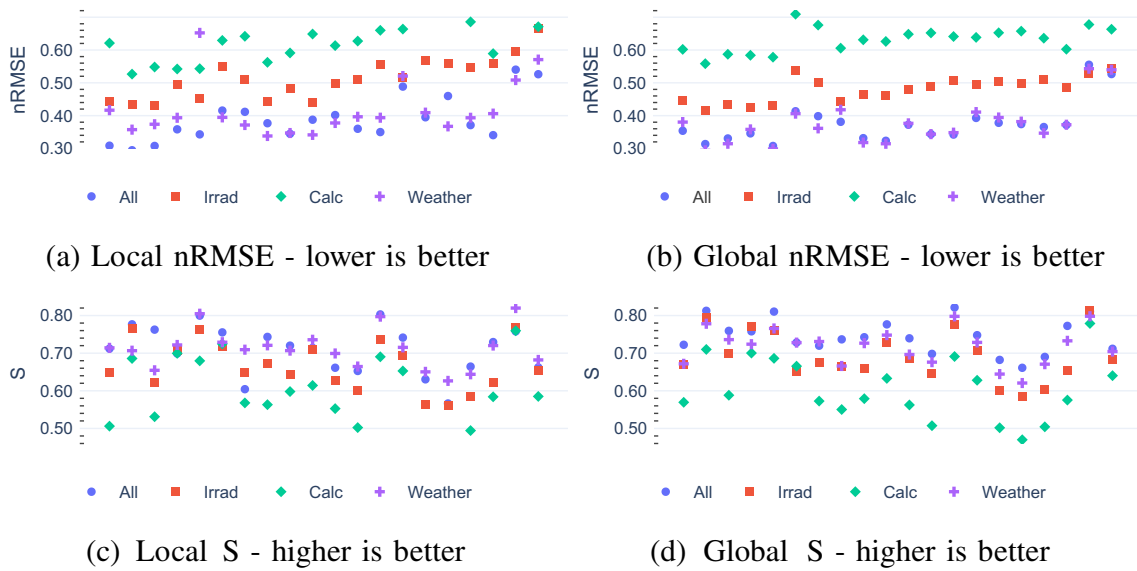


Fig. 8 Distribution of error per AOI for each of the inputs feature groups

Table 10 Average Rankings (Friedman) of the different inputs at steps one and six

		Local		STEP 6		Global		STEP 6	
		STEP 1		Rank	<i>p</i>	Rank	<i>p</i>	STEP 1	
nRMSE	All	1.7	-	1.3	-	2.5	0.001	1.3	-
	Irradiance	2.1	0.327	3.1	0.000	1.1	-	3.0	0.000
	Calc	3.9	0.000	3.8	0.000	4.0	0.000	4.0	0.000
	Weather	2.3	0.283	1.9	0.178	2.4	0.001	1.8	0.221
S	All	2.1	0.327	1.7	-	1.8	0.270	1.3	-
	Irradiance	1.7	-	3.1	0.001	1.4	-	3.0	0.000
	Calc	4.0	0.000	3.5	0.000	4.0	0.000	4.0	0.000
	Weather	2.3	0.222	1.9	0.624	2.9	0.000	1.8	0.221

Bold face is the best-ranked value per step

Table 11 Average of Global and Local results for steps 1-6 for each model

	nRMSE				S			
	Local		Global		Local		Global	
	mean	std	mean	std	mean	std	mean	std
CNN	0.282	0.076	0.205	0.038	0.766	0.050	0.840	0.041
DNN	0.389	0.081	<i>0.376</i>	0.064	0.708	0.067	<i>0.745</i>	0.047
LSTM	0.339	0.066	<i>0.329</i>	0.067	0.744	0.060	<i>0.764</i>	0.054
Transformer	0.335	0.064	<i>0.214</i>	0.039	0.718	0.073	<i>0.834</i>	0.046
Tree	<i>0.334</i>	0.060	0.341	0.063	<i>0.757</i>	0.056	0.753	0.053

In italics is the best of either Local or Global for each method and metric. Bold face is the best overall method for the metric. While in italic is the best result for each model

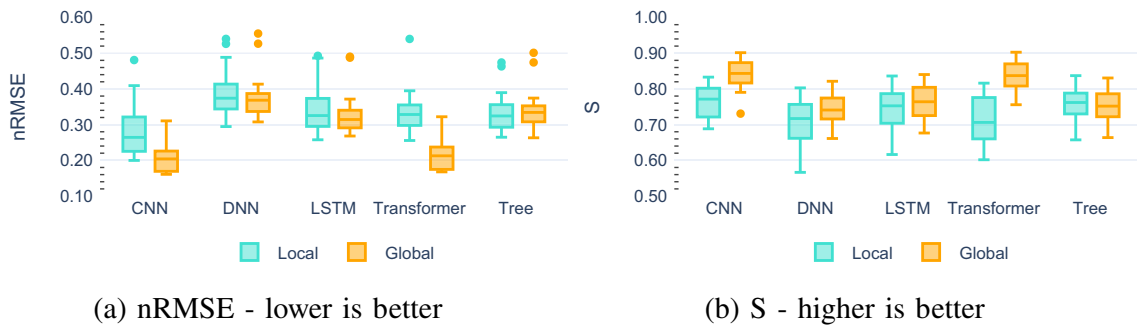


Fig. 9 Distribution of Local and Global errors

improvements. This indicates that both features contain useful information the model can extract. From the per step errors in Fig. 7 it is clear that irradiance features have the largest effect on model performance in the first few steps before. After which, the models' performance decreases towards the model trained on only calculated data, the Calc model. Given a longer forecast horizon, we would expect this trend to continue, as the importance irradiance decreases, until it eventually plateaus in line with the Calc model. The use of just weather data produces a consistent improvement at every step compared to the Calc model. This is unsurprising as, like the calculated data, the relevance of information available for the model to produce the forecasts is the same for all steps.

While the models trained on irradiance data marginally outperform their weather only counterparts at step one ($p < 0.05$). They rapidly reach an inflexion point around step two or three. This drop in the importance of irradiance data explains the switching of rankings at steps one and six in between Irradiance and Weather Table 10. It is clear the models trained on all data were able to extract relevant information from both feature sets; with performance beginning

in line with irradiance before dropping off to be comparable with the Weather model.

Looking at nRMSE errors per AOI in Fig. 8b there are clear performance bands with relatively consistent performance increases for each of the input groups for all locations. However, this trend does not apply to the S as can be seen in Fig. 8d. This would indicate that improvement is not consistent relative to the forecast difficulty of each AOI. Looking at the S per AOI for the static input, given the fairly consistent nRMSE values, the high variance would suggest that some locations are more challenging to predict for than others. This gives more evidence that the inclusion of both weather and irradiance produces better models and explains the drop in the standard deviation of the S from Calc to All we see in both Local and Global flavours.

Given these results, we can see that the use of all features produces the best model. It is worth noting that in both the Local and Global approaches, the use of just weather data appeared to be competitive only being beaten in the first few time steps and having no significant difference in ranking at step six. This is especially evident when we look at nRMSE

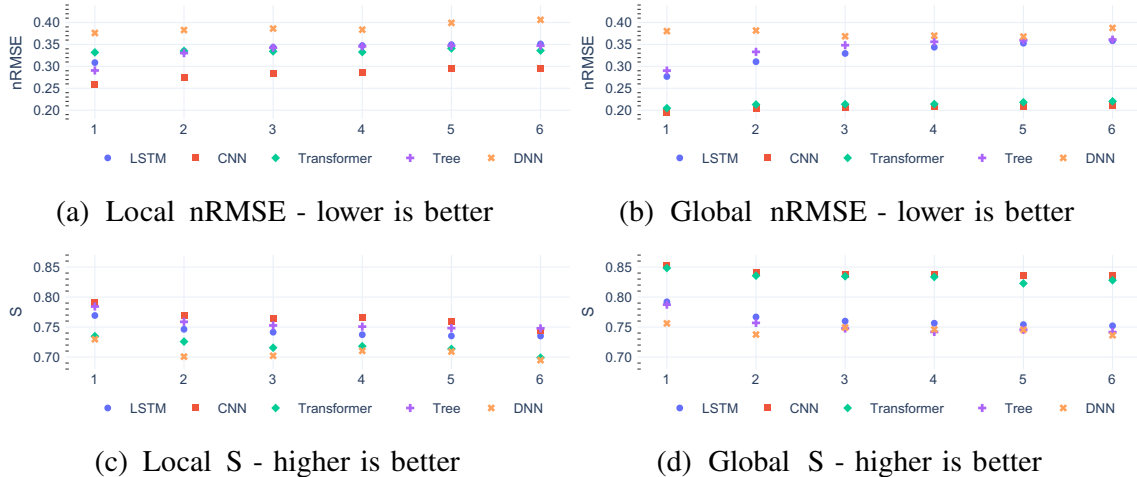


Fig. 10 Distribution of error per step for each of the models comparing local and global

Table 12 Approximate run times to train the models

Model	Local	Global	Compute
CNN	30 mins	12 hours	2x A6000 GPUs
DNN	2 mins	20 mins	2x 2080 GPUs
LSTM	3 mins	30 mins	2x A5000 GPUs
Transformer	1 hour	20 hours	2x A6000 GPUs
Trees	24 CPU mins	500 CPU mins	local 2 cores, global 10 cores

For the trees, this is the time to train 6 trees, one for each forecast step. A large part of the Local training time of the DNN and LSTM (30%) is spent on overheads such as filling caches etc. A single Local training epoch for the DNN took approximately 5 seconds

for the global models where All and Weather have the same score.

5.2 Local vs global

Here we analyse the performance of five the learning models trained in both Local and Global flavours. We aim to compare and understand the effects using Local and Global flavours has on performance for the different models. We trained each of the models using all input features as based on our results in the perverse section we expect doing so to produce the best models.

Table 11 gives the overview of results for each model. Again, we present the average of all AOIs for the full forecast horizon (forecast steps 1 to 6), as well as the standard deviation. Figure 9 show the results for all AOIs by model while and Fig. 10 shows the per forecast step results for each model.

The CNN is the best overall model in both its Local and Global flavours. Interestingly, while the Global Transformer and CNN perform almost identically with both models significantly outperforming all the others; the performance of the Local Transformers drops to be in line with Local DNN. Since the delta between the Local CNN and Transformer is so high it would suggest that the Local Transformer is failing to learn perhaps due to the limited number of samples a single AOI can provide

Overall, from the results in Table 11 it is clear that the Global flavour outperforms the Local for all the DL mod-

els. The per step errors in Fig. 10 show the Global flavour improves results at every time step for the DL models. However, for the Trees, performance is almost identical between the two approaches with the Local flavour presenting a slightly better average. But, at both steps one and six this difference is insignificant ($p > 0.2$).

Another advantage of the Global flavoured DL models, in addition to the improved average error, is a lower variance per AOI. We suspect this increase in consistency is because the Global models are able to extract information from one AOI and apply it to another.

The improved performance of the Global flavour DL models comes with a minimal increase in total computational cost. Since the time complexity for training the models is $O(n)$, sequentially training a Local model for each AOI or a single Global model on all AOIs, will take approximately the same amount of time. Table 12 gives an overview of the approximate training times for each method clearly showing the relationship of $\approx 1 : 20$. While it is possible to easily parallelise training the local flavour using scale out to train multiple AOIs at once². This is not possible for the Global approach as the model needs to be trained on all the data. However, the inability to easily parallelise the Global models is their advantage, and we suspect a reason it outperforms the Local, it sees more data.

We also note that the same time and data constraints are true for the Trees, growing in line with the amount of data used. However, unlike the DL methods, the Global approach does not seem to provide a performance gain for Trees.

5.2.1 Does location affect performance?

As can be seen in Figs. 8 and 9 there is variance in performance between the AOIs. Here we assess if the variation can be explained by an AOIs location. In Fig. 11 we show the error for every time step and AOI plotted against the AOIs latitude. This gives us an indication if there is any correlation

Table 13 Friedman Rankings of the S error for each Global model per step

Model	Step 1	Step 2	Step 3	Step 4	Step 5	Step 6
CNN	1.5	1.5	1.5	1.3	1.2	1.4
DNN	4.9	4.8	4.5	4.2	4.3	4.4
LSTM	3.4	3.3	3.4	3.5	3.7	3.5
Transformer	1.6	1.6	1.6	1.7	1.9	1.6
Tree	3.8	3.9	4.2	4.4	4.0	4.1

The best value is in bold

² All DL Local models were trained using vector parallelism saving significant wall time

between how far North / South an AOI is and its performance relative to the other AOIs.

Looking at the nRMSE there is a correlation between the AOIs error and its latitude with AOIs further north appearing to perform worse. $r = 0.74$ and $r = 0.67$ for Local and Global respectively. However, when comparing the S error, the correlation is not as strong, with Local $r = -0.48$ and Global $r = -0.31$. We believe this is because the nRMSE values are normalised by the mean irradiance of the AOI and locations further north receive less irradiance throughout the year and as such have a lower average exacerbating any forecast error relative to AOIs further south.

Overall, while location likely does play some part in the performance of any given AOI, we feel it is not as significant a factor relative to other factors that may affect model performance at any AOI.

5.2.2 Data used / learning methods

Of the methods that make use of point weather data; DNN, LSTM and Trees, their performance is extremely consistent. This is especially true for the Global models where their performance is almost identical as can be seen in Fig. 10. It is also emphasised in Table 13 showing the Global model rankings at every step. The fact that all three methods appear to perform comparably while the CNNs and Transformers show a significant improvement, suggests there is a limit on the amount of information that can be extracted using point weather. The use of the image data seems to break through the information floor, supported by the fact that the local CNN outperforms all point weather data methods' Global flavour.

We suspect this is due to the fact it receives a richer view of the weather as there is always some distance between the

observation station and the AOI. However, this improved performance comes with a much larger computational cost to train and run the models. Both the CNNs and Transformers take significantly longer to train than the other methods.

5.3 GPH and GPkN

We have shown in Section 5.1, that the use of real-time irradiance can improve accuracy for the first few steps of the predictions. However, until now we have presented results for the perfect case where data has been available for all AOIs. One of the main aims of our paper is to present an understanding of potential solutions for when there is no or limited access to data at the AOI. In Section 3.2 we presented two data models able to produce forecasts for AOIs with limited data, GPH for a lack of historic data and GPkN for a lack of real-time data at the AOI. In this section, we analyse the performance of models trained using these alternate data models compared to their Local and Global counterparts.

Figure 12 shows the distribution of error per AOI for each of the models four flavours, Local, Global, GPH and GPkN. Figure 13 shows the S error for each of the five learning models at every forecast step. While Table 14 shows the Friedman ranking for each method at forecast steps one and six.

From Fig. 12 it is clear that for the DL methods both GPH and GPkN appear to be viable approaches, in most cases outperforming their Local counterparts. For the CNN and Transformers, the GPH models yield results in line with their Global flavours. Looking at the per-step errors in Fig. 13 it is clear they follow the same trend with a number of steps being indistinguishable. This is supported by rankings in Table 14 with Global and GPH consistently ranking in line with one another.

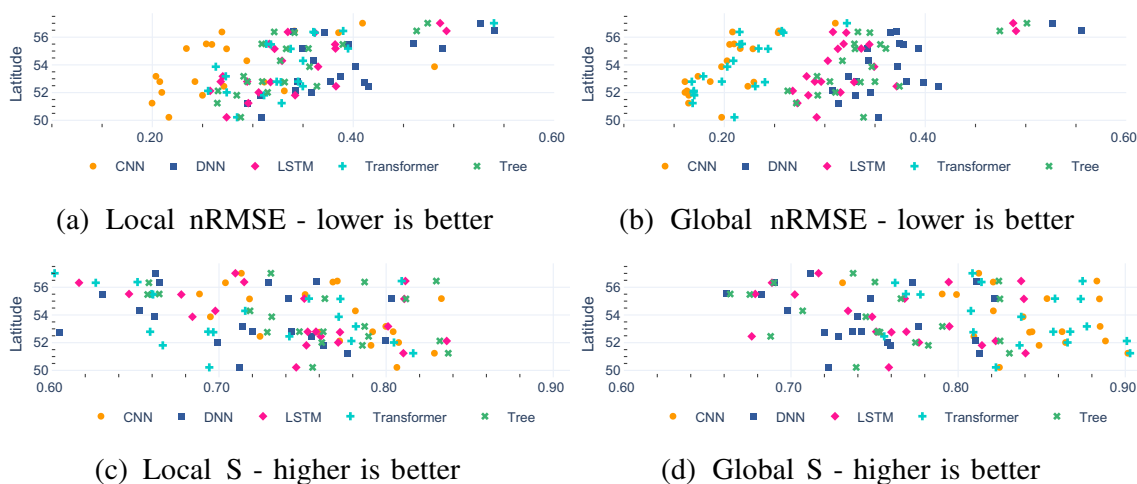


Fig. 11 Error for all forecast steps 1-6 compared to AOI latitude (N/S)

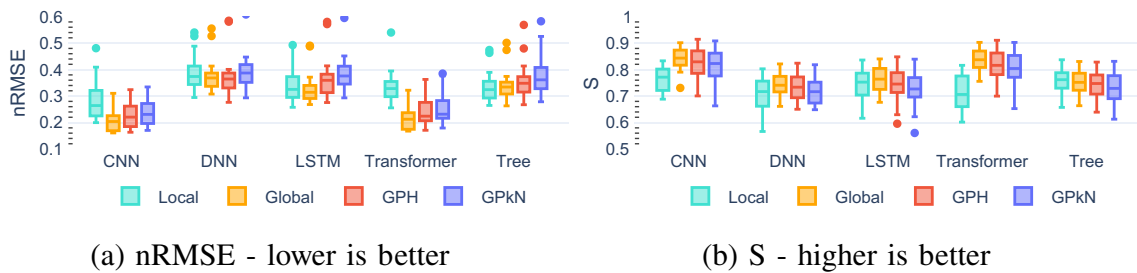


Fig. 12 Overview of Local, Global, GPH and GPkN per AOI. The Box plot shows the average error for each AOI

In the case of GPkN applied to the DL models from Fig. 12 they appear to fit between the Local and Global flavours. Looking at the per-step error in Fig. 13 it is likely this is caused by the GPkN under-performing relative to even the Local approach for the first few forecast steps. By the end of the forecast horizon, the GPkN improves but only marginally worse than the GPH. We can observe this effect in the rankings, GPkN ranks last at step one but by step six it usually ranks closer to the GPH.

The Trees seem to buck this trend. As already noted in Section 5.2, their performance for Local and Global data models has very little difference. Interestingly both the GPH and GPkN versions seem to perform comparably and worse than both the Local and Global approaches. This indicates that the Trees fail to generalise to unseen locations. This could suggest that they overly rely on correlations learned

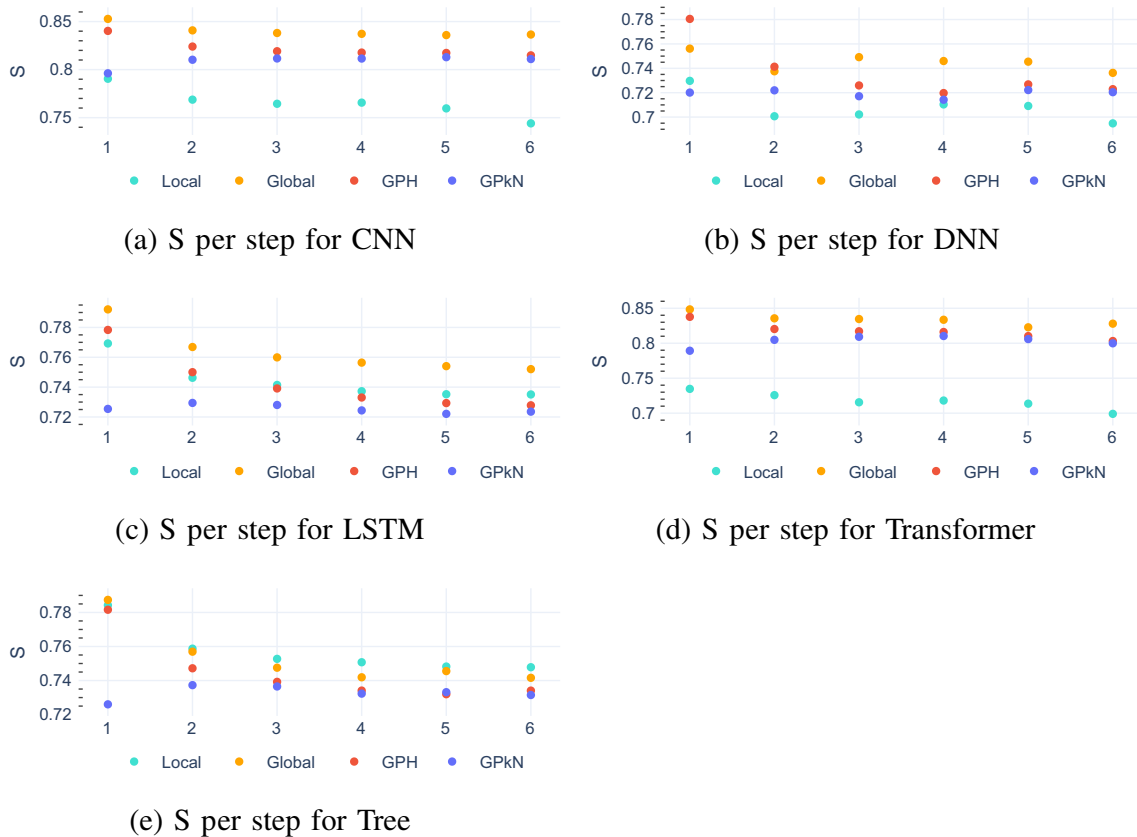


Fig. 13 S error per forecast step for the Local, Global, GPH and GPkN approach for at steps 1-6 for each of the 4 models

Table 14 Friedman rankings of the different data models at steps one and six for both error metrics

		Step 1				Step 6			
		RMSE		S		RMSE		S	
		Ranking	<i>p</i>	Ranking	<i>p</i>	Ranking	<i>p</i>	Ranking	<i>p</i>
CNN	GPH	2.2	0.005	2.2	0.017	1.8	0.391	2.2	0.017
	Global	1.0	-	1.4	-	1.5	-	1.4	-
	GpKN	3.5	0.000	2.6	0.025	3.4	0.000	2.6	0.025
	Local	3.4	0.000	3.8	0.050	3.4	0.000	3.8	0.050
DNN	GPH	1.1	-	2.4	0.050	1.4	-	2.4	0.066
	Global	3.2	0.017	2.1	-	2.4	0.020	1.6	-
	GpKN	2.9	0.025	2.9	0.017	3.1	0.000	2.7	0.014
	Local	2.8	0.050	2.7	0.025	3.2	0.000	3.4	0.000
LSTM	GPH	2.4	0.003	3.1	0.001	2.2	0.050	2.6	0.020
	Global	1.2	-	1.7	-	1.4	-	1.6	-
	GpKN	3.9	0.000	3.4	0.000	3.6	0.000	3.3	0.000
	Local	2.6	0.002	1.8	0.806	2.8	0.001	2.6	0.020
Transformer	GPH	1.9	0.111	2.3	0.050	1.8	0.220	2.1	0.111
	Global	1.2	-	1.5	-	1.3	-	1.5	-
	GpKN	3.1	0.000	2.5	0.020	3.1	0.000	2.5	0.020
	Local	3.9	0.000	3.8	0.000	3.9	0.000	4.0	0.000
Tree	GPH	2.5	0.014	2.5	0.132	3.3	0.000	2.9	0.259
	Global	1.5	-	1.7	-	1.9	0.391	2.2	-
	GpKN	4.0	0.000	3.7	0.000	3.3	0.000	2.8	0.259
	Local	2.0	0.270	2.2	0.270	1.6	-	2.2	0.903

The best-ranked method is in bold

from the real-time irradiance of the AOI seen to make their predictions.

Leveraging data from alternative AOIs to produce predictions for AOIs with none. Both GPH and GPKn appeared to be viable methods for generating forecasts for AOIs where there is limited, or no, data available.

5.3.1 Another look at input features

Both the GPH and GPKn data models enable forecasts for AOI without historic or direct access to data. However, since all the models use some value of observed irradiance, there is still a dependency on real-time data. This is not always possible to access. In this section, we further analyse the effects the use of real-time irradiance has on performance compared to just using weather data. We compare the results from the various models trained both with and without real-time irradiance as an input feature. We specifically focus on the results from the Global and GPKn modes. GPH was omitted as without irradiance as an input feature the results are the same as GPKn.

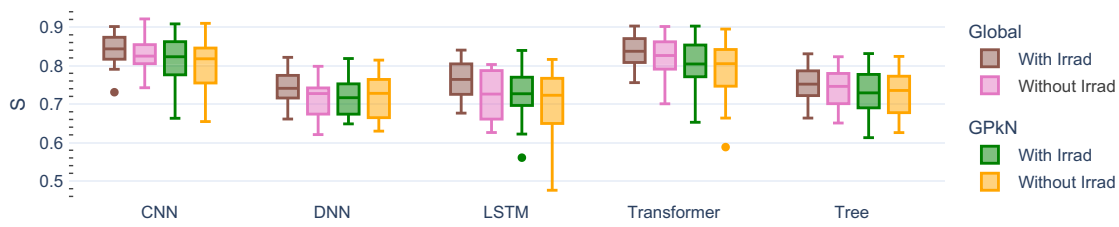
Figure 14 shows the average S per AOI and per step of the Global and GPKn models trained both with and without real-time irradiance as an input feature. In the case of Global models, we clearly see the inclusion of real-time irradiance

improves performance for all the models. This is unsurprising as we have already shown in Section 5.1 that the use of irradiance helps Global models. However, in the case of GPKn where each AOI does not have access to real-time and instead uses a substitute value, the benefit is not as clear.

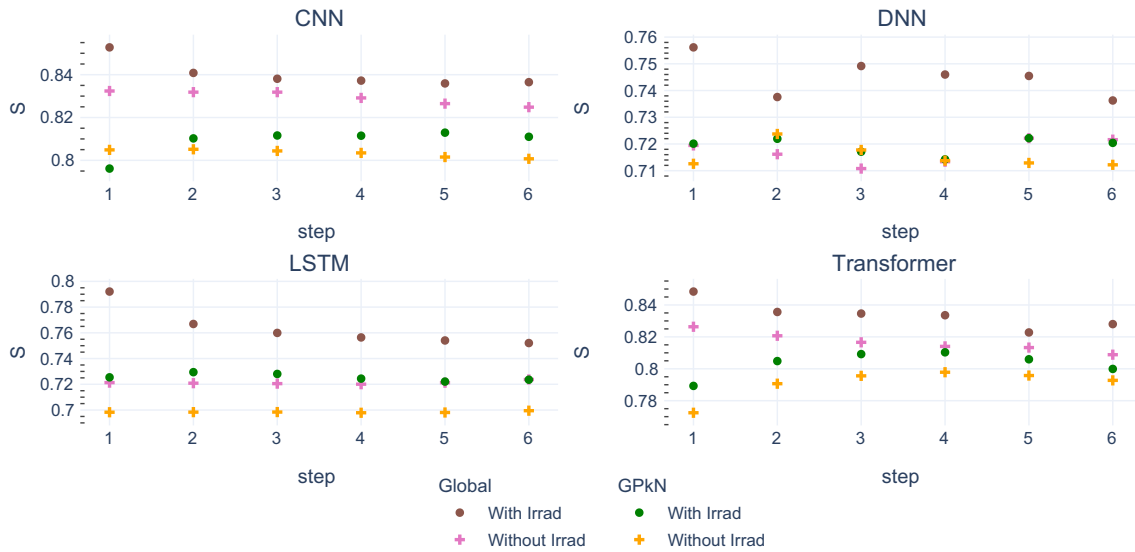
Looking at the perstep errors, the inclusion of irradiance for the GPKn flavour of the CNN has an adverse effect on performance. However, this was not significant $p = 0.36$. Further analysis of all GPKn models this trend continued, unlike the Global models the difference when using irradiance is not statistically significant. In other words, when operating in the worst-case scenario of GPKn the inclusion of irradiance has no real benefit.

We suspect this is due to a poor correlation between the true real-time irradiance used by the Global and the substituted value used in GPKn. With the average distance to the substituted AOI being 68km this is unsurprising. If the substituted values were closer we would expect to see the performance improve.

As we observed in Section 5.1 for Global models, where the real-time irradiance is sourced from the AOI, its inclusion improves model performance for the first few forecast steps. However, for GPKn, when irradiance is not sourced from the AOI, the use of irradiance is of limited value and in some cases may even hinder model performance. As such, when



(a) Distribution of the average S for every AOI



(b) Average S per step

Fig. 14 A comparison of Global and GPKn for all the ML methods with and without irradiance as in input used as an input feature

using GPKn models care must be taken to ensure a correlation of irradiance between the target AOI and the substituted AOI.

5.4 Results summary

When working with multiple AOIs, the Global flavour is better for DL based methods, the Local flavour only makes sense when the ability to parallelise training is worth the cost in performance. While they take longer to train, they produce better and more consistent results. Additionally, in the case of the DL approaches if new data become available they could potentially be further refined by training the existing model on the new data, transfer learning. In the case of the Trees, there is a minimal performance gain compared to using the Local flavour.

Although the Local and Global flavours of all learning models were able to perform well, their need for a complete dataset can limit their usefulness in the real world. We have shown through the use of GPH that the DL models can gen-

eralise well across locations working effectively for unseen locations.

Overall the models benefit from the use of real-time irradiance. More generally it would seem that the inclusion of real-time irradiance can improve overall performance for an AOI where it is available, regardless of the data and training model used. However, its advantage over just using weather data seems to be limited to the first few forecast steps, and for forecasts with horizons longer than a few hours, its use is not as necessary. We have shown it is possible to uncouple irradiance observations from forecasting models, although with marginally reduced performance.

We have also observed that the use of richer weather data, such as that provided by satellite imagery, leads to a significant performance improvement. While there is a larger computational cost compared to the models using point weather data, it is worth it. Both the CNN and Transformer models in the global flavour drastically outperformed their DNN, LSTM and Tree counterparts.

6 Conclusion

In this work, we have explored various techniques for building irradiance forecasting models. We used a number of standard ML methods, DNNs, CNNs, LSTMs, Transformers and Trees. Each trained using four data models: Local, Global, Global Plant Holdout and Global Plant kNear. The Local approach trains a model per location while the Global approach combines data from all locations and trains a single model. GPH and GPkN are extensions to the Global approach used to circumvent data dependency issues that may occur at training time and when the model is in production. GPH model generates forecasts for locations without historic data while GPkN circumvents any real-time data dependency by substituting values from nearby locations. Furthermore, we analysed the effects the use of different input features can have, specifically; real-time irradiance and weather data. We also explored different weather formats, point-based and satellite data.

Experimentally, we have shown that the Global approach and its extensions are superior to the Local. While computationally more expensive to train, utilising all sequences to learn, the single Global model consistently outperformed its local counterpart. Furthermore, the Global approach can be utilised to generate forecasts for locations with limited historical data.

Our experiments have shown that the use of real-time irradiance can improve forecasts for the first few steps, however, after 2-3 hours its importance diminishes, and weather data is key. When attempting to forecast for locations without direct access to real-time data, while it is possible to substitute irradiance values from other locations care must be taken. The greater the distance between the two locations, the weaker their irradiance will correlate, and performance will be negatively impacted.

Additionally, our experimental results have shown that the use of satellite images works very well. While the DNN, LSTM and Trees perform in line with each other, the CNN and Transformers using satellite imagery consistently outperform all of them. In fact, the CNN operating in its worst case, GPkN, presents an improvement over its Global point weather counterparts. While in practice, access to these kinds of forecasts may be challenging; this result would strongly suggest that the use of richer weather data, over single-point data, can significantly improve forecast accuracy.

The proposed models are capable of being integrated into a planning and optimisation system for use in the energy market. However, further exploration of the effect richer weather data has on performance is needed. We plan to work to create mixed modal models that can combine satellite and point weather data. We also plan to develop methods to deal with increased and variable temporal resolution data, in an effort

to better capture intra-hour fluctuations. There is also scope for work to address issues such as: How best to deal with sets of AOIs that are extremely distributed such as locations in both the UK and Australia; GPkN can be improved by using more robust imputation methods, perhaps by combining data from multiple locations rather than just using the closest location. Additionally, while GPH and GPkN work for AOIs known to not have data, work is needed to develop methods to handle transient missing data due to temporary outages or network errors.

Acknowledgements The authors would like to thank Elastacloud for providing access to data and computing resources. We gratefully acknowledge the support of NVIDIA Corporation with the donation of the Titan Xp GPU used for this research. T. Cargan holds a studentship funded by EPSRC and The University of Nottingham. I. Triguero is funded by a Maria Zambrano Senior Fellowship at the University of Granada. This work is also supported by the Spanish projects A-TIC-434-UGR20 and PID2020-119478GB-I00.

Author Contributions Conceptualization: Timothy R. Cargan; Methodology: Timothy R. Cargan; Formal analysis and investigation: Timothy R. Cargan; Writing - original draft preparation: Timothy R. Cargan; Writing - review and editing: Timothy R. Cargan, Isaac Triguero, Dario Landa-Silva; Supervision: Isaac Triguero, Dario Landa-Silva

Funding I. Triguero's work is supported by projects A-TIC-434-UGR20 and PID2020-119478GB-I00. I. Triguero is currently enjoying a Maria Zambrano fellowship at the University of Granada.

Data Availability The raw image data can be found at [12]; The raw irradiance data can be found [43]; The weather data was sourced from [8]. The processed datasets are available from the corresponding author on reasonable request.

Declarations

Conflict of interest/Competing interests The authors have no competing interests to declare that are relevant to the content of this article.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

References

1. Reno M, Hansen C, Stein J (2012). Global horizontal irradiance clear sky models: implementation and analysis. <https://doi.org/10.2172/1039404>

2. Edward Hughes IMSJohn Hiley, Brown K (2008) Electrical energy systems. In: Hughes Electrical and Electronic Technology. Pearson/Prentice Hall, chap 39, p 824–828
3. Elexon (2023) Imbalance pricing. <https://www.elexon.co.uk/operations-settlement/balancing-and-settlement/imbalance-pricing/>
4. Jones LE (2014) Renewable Energy Integration: Practical Management of Variability, Uncertainty, and Flexibility in Power Grids. Elsevier Science
5. Yang B, Zhu T, Cao P et al (2023) Classification and summarization of solar irradiance and power forecasting methods: A thorough review. CSEE Journal of Power and Energy Systems 9(3):978–995. <https://doi.org/10.17775/CSEEJPES.2020.04930>
6. Lago J, Brabandere KD, Ridder FD et al (2018) Short-term forecasting of solar irradiance without local telemetry: A generalized model using satellite data. Sol Energy 173:566–577. <https://doi.org/10.1016/j.solener.2018.07.050>
7. Zheng J, Zhang H, Dai Y et al (2020) Time series prediction for output of multi-region solar power plants. Appl Energy 257:114001. <https://doi.org/10.1016/j.apenergy.2019.114001>
8. weatherbit (2023) Data source - weatherbit. <https://www.weatherbit.io>
9. openweathermap (2023) Data source - openweathermap. <https://openweathermap.org>
10. Rodríguez-Abreo O, Cervantes I (2022) The role of input selection and climate pre-classification on the performance of neural networks irradiance models. Appl Soft Comput 130:109643. <https://doi.org/10.1016/j.asoc.2022.109643>
11. Tsai WC, Tu CS, Hong CM et al (2023) A review of state-of-the-art and short-term forecasting models for solar pv power generation. Energies 16:5436. <https://doi.org/10.3390/en16145436>
12. EUMETSAT (2023) High rate seviri level 1.5 image data - msg - 0 degree. <https://navigator.eumetsat.int/product/EO:EUM:DAT:MSG:HRSEVIRI>
13. Gallo R, Castangia M, Macii A et al (2022) Solar radiation forecasting with deep learning techniques integrating geostationary satellite images. Eng Appl Artif Intell 116:105493. <https://doi.org/10.1016/j.engappai.2022.105493>
14. Mercier TM, Rahman T, Sabet A (2023) Solar irradiance anticipative transformer. IEEE, pp 2065–2074. <https://doi.org/10.1109/CVPRW59228.2023.00200>
15. Montero-Manso P, Hyndman RJ (2021) Principles and algorithms for forecasting groups of time series: Locality and globality. Int J Forecast 37:1632–1653. <https://doi.org/10.1016/j.ijforecast.2021.03.004>
16. Semenoglou AA, Spiliotis E, Makridakis S et al (2021) Investigating the accuracy of cross-learning time series forecasting methods. Int J Forecast 37:1072–1084. <https://doi.org/10.1016/j.ijforecast.2020.11.009>
17. Ahmed R, Sreeram V, Mishra Y et al (2020) A review and evaluation of the state-of-the-art in PV solar power forecasting: Techniques and optimization. Renew Sustain Energy Rev 124:109792. <https://doi.org/10.1016/j.rser.2020.109792>
18. García-Cuesta E, Aler R, del Pózo-Vázquez D et al (2022) A combination of supervised dimensionality reduction and learning methods to forecast solar radiation. Appl Intell. <https://doi.org/10.1007/s10489-022-04175-y>
19. Kim B, Suh D (2020) A hybrid spatio-temporal prediction model for solar photovoltaic generation using numerical weather data and satellite images. Remote Sensing 12:3706. <https://doi.org/10.3390/rs12223706>
20. Niu D, Wang K, Sun L et al (2020) Short-term photovoltaic power generation forecasting based on random forest feature selection and ceemd: A case study. Appl Soft Comput 93:106389. <https://doi.org/10.1016/j.asoc.2020.106389>
21. Goodfellow I, Bengio Y, Courville A (2016) Deep Learning. MIT Press
22. Lara-Benítez P, Carranza-García M, Luna-Romera JM et al (2023) Short-term solar irradiance forecasting in streaming with deep learning. Neurocomputing 546:126312. <https://doi.org/10.1016/j.neucom.2023.126312>
23. Cesar LB, e Silva RA, Ángel Manso Callejo M et al (2022) Review on spatio-temporal solar forecasting methods driven by in situ measurements or their combination with satellite and numerical weather prediction (nwp) estimates. Energies 15:4341. <https://doi.org/10.3390/en15124341>
24. Lipton ZC, Berkowitz J, Elkan C (2015) A critical review of recurrent neural networks for sequence learning. CoRR abs/1506.00019. <https://arxiv.org/abs/arXiv:1506.00019>
25. Liu S, Ji H, Wang MC (2019) Nonpooling convolutional neural network forecasting for seasonal time series with trends. IEEE Trans Neural Netw Learn Syst pp 1–10. <https://doi.org/10.1109/TNNLS.2019.2934110>
26. Passalis N, Tefas A, Kannianen J et al (2019) Deep Adaptive Input Normalization for Time Series Forecasting. IEEE Transactions on Neural Networks and Learning Systems pp 1–6. <https://doi.org/10.1109/TNNLS.2019.2944933>
27. Tran DT, Iosifidis A, Kannianen J et al (2019) Temporal Attention-Augmented Bilinear Network for Financial Time-Series Data Analysis. IEEE Transactions on Neural Networks and Learning Systems 30(5):1407–1418. <https://doi.org/10.1109/TNNLS.2018.2869225>
28. Alassafi MO, Jarrah M, Alotaibi R (2022) Time series predicting of covid-19 based on deep learning. Neurocomputing 468:335–344. <https://doi.org/10.1016/j.neucom.2021.10.035>
29. Weerakody PB, Wong KW, Wang G et al (2021) A review of irregular time series data handling with gated recurrent neural networks. Neurocomputing 441:161–178. <https://doi.org/10.1016/j.neucom.2021.02.046>
30. Bishop CM (2006) Pattern Recognition and Machine Learning (Information Science and Statistics). Springer-Verlag, Berlin, Heidelberg
31. Chodakowska E, Nazarko J, Nazarko Łukasz et al (2023) Arima models in solar radiation forecasting in different geographic locations. Energies 16:5029. <https://doi.org/10.3390/en16135029>
32. Greff K, Srivastava RK, Koutník J et al (2017) LSTM: A Search Space Odyssey. IEEE Transactions on Neural Networks and Learning Systems 28(10):2222–2232. <https://doi.org/10.1109/TNNLS.2016.2582924>
33. Akhter MN, Mekhilef S, Mokhlis H et al (2022) An hour-ahead PV power forecasting method based on an RNN-LSTM model for three different PV plants. Energies 15:2243. <https://doi.org/10.3390/en15062243>
34. Pan C, Tan J, Feng D (2021) Prediction intervals estimation of solar generation based on gated recurrent unit and kernel density estimation. Neurocomputing 453:552–562. <https://doi.org/10.1016/j.neucom.2020.10.027>
35. Vaswani A, Shazeer N, Parmar N et al (2017) Attention is all you need. vol 30. Curran Associates, Inc. <https://proceedings.neurips.cc/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf>
36. Dosovitskiy A, Beyer L, Kolesnikov A et al (2021) An image is worth 16x16 words: Transformers for image recognition at scale. In: International Conference on Learning Representations. URL <https://openreview.net/forum?id=YicbFdNTTy>
37. Demir A, Gutierrez LF, Namin AS et al (2022) Solar irradiance prediction using transformer-based machine learning models. IEEE, pp 2833–2840. <https://doi.org/10.1109/BigData55660.2022.10020615>

38. Pospíchal J, Kubovčík M, Luptáková ID (2022) Solar irradiance forecasting with transformer model. *Appl Sci* 12:8852. <https://doi.org/10.3390/app12178852>
39. Gao H, Liu M (2022) Short-term solar irradiance prediction from sky images with a clear sky model. *IEEE*, pp 3074–3082. <https://doi.org/10.1109/WACV51458.2022.00313>
40. Zambrano AF, Giraldo LF (2020) Solar irradiance forecasting models without on-site training measurements. *Renewable Energy* 152:557–566. <https://doi.org/10.1016/j.renene.2020.01.092>
41. Bottieau J, Grève ZD, Piraux T et al (2022) A cross-learning approach for cold-start forecasting of residential photovoltaic generation. *Electric Power Systems Research* 212:108415. <https://doi.org/10.1016/j.epr.2022.108415>
42. Triguero I, Galar M (2023) *Large-Scale Data Analytics with Python and Spark: A Hands-on Guide to Implementing Machine Learning Solutions*. Cambridge University Press. <https://doi.org/10.1017/9781009318242>
43. Office Met (2022) Midas open: Uk hourly solar radiation data, v202207. NERC EDS Centre for Environmental Data Analysis. <https://doi.org/10.5285/e3a7f3336ff8464f9ae6534a8e8676e5>
44. Holmgren W, Hansen C, Mikofski M (2018) pvlib python: a python package for modeling solar energy systems. *J Open Source Softw* 3:884. <https://doi.org/10.21105/joss.00884>
45. Breiman L (2001) Random forests. *Mach Learn* 45. <https://doi.org/10.1023/A:1010933404324>
46. Guillaume-Bert M, Bruch S, Stotz R et al (2023) Yggdrasil decision forests: A fast and extensible decision forests library. In: *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, KDD 2023, Long Beach, CA, USA, August 6–10, 2023*, pp 4068–4077. <https://doi.org/10.1145/3580305.3599933>
47. Vens C, Struyf J, Schietgat L et al (2008) Decision trees for hierarchical multi-label classification. *Mach Learn* 73:185–214. <https://doi.org/10.1007/s10994-008-5077-3>
48. Basgalupp M, Cerri R, Schietgat L et al (2021) Beyond global and local multi-target learning. *Inf Sci* 579:508–524. <https://doi.org/10.1016/j.ins.2021.08.022>
49. Bergmeir C, Hyndman RJ, Koo B (2018) A note on the validity of cross-validation for evaluating autoregressive time series prediction. *Computational Statistics & Data Analysis* 120:70–83. <https://doi.org/10.1016/j.csda.2017.11.003>
50. Bergmeir C, Benítez JM (2012) On the use of cross-validation for time series predictor evaluation. *Inf Sci* 191:192–213. <https://doi.org/10.1016/j.ins.2011.12.028>
51. Hyndman RJ, Koehler AB (2006) Another look at measures of forecast accuracy. *Int J Forecast* 22(4):679–688. <https://doi.org/10.1016/j.ijforecast.2006.03.001>
52. Marquez R, Coimbra CFM (2012) Proposed metric for evaluation of solar forecasting models. *J Sol Energy Eng* 135(1). <https://doi.org/10.1115/1.4007496>
53. Sheskin DJ (2003) *Handbook of parametric and nonparametric statistical procedures*. Chapman and hall/CRC
54. Demšar J (2006) Statistical comparisons of classifiers over multiple data sets. *The Journal of Machine Learning Research* 7:1–30
55. Garcia S, Herrera F (2008) An extension on “statistical comparisons of classifiers over multiple data sets” for all pairwise comparisons. *J Mach Learn Res* 9(12)
56. Hodges JL Jr, Lehmann EL (1962) Rank methods for combination of independent experiments in analysis of variance. *Ann Math Stat* 33(2):482–497
57. Holm S (1979) A simple sequentially rejective multiple test procedure. *Scand J Stat* pp 65–70

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.