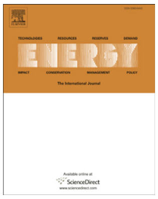
Energy 240 (2022) 122812

Contents lists available at ScienceDirect

Energy

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



Improved solar photovoltaic energy generation forecast using deep learning-based ensemble stacking approach

Waqas Khan[*], Shalika Walker, Wim Zeiler

Eindhoven University of Technology, Netherlands

A R T I C L E I N F O

Article history:
Received 16 July 2021
Received in revised form
30 November 2021
Accepted 2 December 2021
Available online 4 December 2021

Keywords:
Solar energy forecasting
Uncertainty
Machine learning
Deep learning
Stacking
Ensemble learning

A B S T R A C T

An accurate solar energy forecast is of utmost importance to allow a higher level of integration of renewable energy into the controls of the existing electricity grid. With the availability of data in unprecedented granularities, there is an opportunity to use data-driven algorithms for improved prediction of solar generation. In this paper, an improved generally applicable stacked ensemble algorithm (DSE-XGB) is proposed utilizing two deep learning algorithms namely artificial neural network (ANN) and long short-term memory (LSTM) as base models for solar energy forecast. The predictions from the base models are integrated using an extreme gradient boosting algorithm to enhance the accuracy of the solar PV generation forecast. The proposed model was evaluated on four different solar generation datasets to provide a comprehensive assessment. Additionally, the shapely additive explanation framework was utilized in this study to provide a deeper insight into the learning mechanism of the algorithm. The performance of the proposed model was evaluated by comparing the prediction results with individual ANN, LSTM, and Bagging. The proposed DSE-XGB method exhibits the best combination of consistency and stability on different case studies irrespective of the weather variations and demonstrates an improvement in R^2 value of 10%–12% to other models.

1. Introduction

With the increasing energy demand, the world is moving towards alternative renewable energy resources to reduce greenhouse gas emissions [1]. The high penetration of renewables in the power system provides many environmental and economic benefits, but with the characteristic of intermittency and variability, renewable energy brings challenges for the reliability and safe operation of the power system. Due to the advances in PV technology, solar energy has gained considerable importance in recent years. The energy conversion efficiency is increased while the costs for panels installation and electricity generation have fallen significantly [1]. Considering the abundant availability of resources and cost competitiveness solar PV is anticipated to continue the overall renewable energy growth over the next decade [2].

Accurate forecasting techniques have become important for the stable and safe integration of renewable energy resources into the existing power grid [2] and the better alignment of supply and demand. Most importantly, as elements associated with the energy grid electrifies (e.g.: introduction of heat pumps), the level of energy self-sufficiency achieved by the buildings and neighbourhoods are found to be inadequate [3]. Accurate forecasting is the degree of closeness of the predicted value of the generation of PV panels to the actual (true) value. The forecast of solar PV plays an important role in the evolving energy roadmap for congestion management, estimating the reserves, management of storage, the energy exchange between buildings, and grid integration [4]. Nevertheless, the integration of smart meters and the availability of data has opened new opportunities to use data-driven machine learning and deep learning algorithms for the improved prediction of PV generation. Diverse research studies have tried to integrate such data-driven algorithms in the recent past.

1.1. Forecasting methods based on machine learning

Several forecasting methods [5–7] are proposed in the literature based on machine learning for solar PV generation is shown in Table 1. However, there is no single method capable of accurately performing on multiple case studies. Switching from one solar PV plant (dataset) to another proposes various types of variabilities for

[*] Corresponding author.
E-mail address: w.khan@tue.nl (W. Khan).

https://doi.org/10.1016/j.energy.2021.122812

Nomenclature

Abbreviations

Adaboost	Adaptive boosting
AE	Autoencoders
Agg	Aggregated
ANN	Artificial neural network
ARIMA	Autoregressive moving average
BP	Back Propagation
CNN	Convolutional neural network
CV	Cross-validation
DBN	Deep belief networks
DSE	Deep stacked ensembles
DSE-XGB	Deep stacked ensemble-XGBoost
DT	Decision trees
GB	Gradient boosting
GHI	Global horizontal irradiance
KNN	K-Nearest Neighbour
LSTM	Long short-term memory
MAE	Mean absolute error
MAPE	Mean Absolute Percentage Error
NRMSE	Normalised Root Mean Square
NRMSE	Normalised Root mean square error
NWP	Numerical Weather Parameters
PV	Photovoltaic
RBM	Restricted Boltzman machine
RH	Relative humidity
RMSE	Root mean square error
RNN	Recurrent neural network
SHAP	SHapley Additive exPlanations
SOM	Self-Organizing Maps
SVM	Support Vector Machines
Temp	Temperature
XGBoost	Extreme gradient boosting

Table 1
Results of solar forecasting algorithms with different time horizons.

Publications	Models	Time Horizon	Error Metrics	Results
Shi et al. [5]	Solar PV Power Output	24 h-ahead	MRE	8.64
Chen et al. [24]	SOM's, ANN	24 h-ahead	MAPE	Sunny days (10.8%), cloudy days (11.05), rainy days (38.104)
Lin et al. [27]	Pattern sequence neural network	24 h-ahead	MAE, RMSE	77.37, 106.33
Gensler et al. [32]	Autoencoder, LSTM	24 h-ahead	RMSE, MAE	0.713, 0.0366
Vaz et al. [26]	ANN	Up to 1-h ahead	NRMSE	9% to 25%
Qing and Niu [33]	LSTM	1 h ahead	RMSE	18.34%
Wang et al. [28]	CNN, ANN, LSTM	24 h-ahead	RMSE, MAE, MAPE	0.343–1.434, 0.126–0.628, 0.022–0.112
Caroline et al. [14]	XGBoost	1-6 h-ahead	NRMSE	0.09–0.12

machine learning models. The outcome of different algorithms using the same weather and power data or the same method with different internal hyperparameter settings can lead to different predictions outcome. The models are reported to have volatility issues, i.e., a small change in input data may result in substantial variations in the prediction values that affect the reliability of models [8]. The modelling may not reproduce the same results on a new dataset because of uncertainties in the PV module properties, data collection errors, system output performance, and long-term effects [9].

To overcome these challenges, considerable research [10–12] has been done to generate accurate solar PV forecasts. Out of these, the use of machine learning ensemble algorithms such as Extreme Gradient Boosting (XGBoost) [13], Gradient Boosting Regression Trees (GBRT) [14], and Random Forest (RF) has shown promising results compared to individual models. The ensemble approaches demonstrated that they are more stable and can decrease the associated uncertainty link to the input data [15]. For example, Usman et al. [16] developed a framework to evaluate various machine learning models and feature selection methods for short-term PV power forecasting. They acknowledged that the XGBoost method outperforms individual machine learning methods. The authors in Ref. [17] presented the difference in the accuracy of the PV power forecast for varying forecast horizons. They concluded that the accuracy of the models decreases with the increasing forecasting horizon. As PV generation is well characterized with high variability periods on partially cloudy days and lower during sunny days, the weather information provided for a specific site is not always precise enough to correctly model these time periods. To overcome this problem Andrade et al. [18] developed a forecasting framework by combining XGboost with feature engineering techniques. The authors suggested that deep learning techniques are more appealing to pursue in combination with proper feature management compared to machine learning.

1.2. Forecasting methods based on deep learning

Deep learning is gaining attention in recent years due to its ability to handle complex data and the advancements in computational power [19]. Contrary to machine learning models deep learning models are more stable on new datasets as changing the input slightly won't affect the original hypothesis learned by the algorithm. Various architectures of ANN's have been applied successfully to forecast renewable energy from a simple ANN network to more complex models like an autoencoder [20], self-organizing maps [21], and LSTM [22] networks. Autoencoder and self-organizing maps are unsupervised deep learning algorithms for classification and encoding of the data whereas LSTM networks are used for time series data forecast as a supervised learning algorithm.

Abuella et all [23] employed an ANN for producing solar power forecasts. They utilized sensitivity analysis to select the best input variable and compared the results with multiple linear regression. Chen et al. [24] presented an advanced statistical method for a 24 h ahead solar power forecast based on ANN. Almonacid et al. [25] presented a method based on ANN to predict 1 h ahead power of solar PV. The method employed dynamic ANN to predict the air temperature and global irradiance that is then used as an input for another ANN to predict the solar PV output. Vaz et al. [26] utilized ANN with measurements from neighbouring PV systems as inputs along with the weather parameters for solar power forecast. They demonstrated that with increasing the time horizon of the forecast, model results decayed and the RMSE increased to 24%. Yang et al. [27] combined pattern sequences extraction algorithm with neural networks for a day ahead forecast for solar PV power. The main idea is to build a separate prediction model for each pattern sequence type. The model is limited only to day-ahead forecasting due to the dependency on the extracted features. Wang et al. [28] compared three deep learning networks for solar power forecasting and provided suggestions for choosing the most suitable network in practical application. They concluded that that deep learning is very helpful for improving the accuracy of PV power prediction.

1.3. Ensemble learning-based forecasting models

Ensemble learning refers to a machine learning technique where multiple base learners are trained and their output is combined, to solve the same problem. The main principle is that the combined output of the base learners should have better accuracy overall, than any individual learner. Several theoretical and empirical studies have established that ensemble learning performs better than single models. Base learners can be any trained machine learning model such as linear regression, KNN, SVM, or ANN's. Ensemble methods can either use a single algorithm to produce homogeneous base learners or a combination of multiple algorithms to produce heterogeneous learners [29].

The authors in Ref. [30] proposed a multi-model ensemble based on statistical models, SVM, and ANN using various numerical weather parameters (NWP). The input NWP parameters were obtained from the weather research forecasting model (WRF) and globally integrated forecasting system model (IFS) to compare the effects of selecting input weather parameters from different providers or models. It was established that the outcome varies using a similar machine learning model with different NWP inputs. The authors in Ref. [31] presented a new ensemble model based on ANN and XGBoost and integrating their output using ridge regression. They evaluated the model performance by comparing the prediction results with different models, such as support vector regression, random forest, extreme gradient boosting forest, and deep neural networks. They also concluded that ensemble models are more accurate and stable compared to individual modelling.

1.4. Overcoming the current limitation of forecasting methods

Most of the previous research in deep ensemble learning has treated Solar PV generation only as a regression task [10–31] by only using artificial neural network models and statistical models at the base level. The dynamic behaviour of solar PV time series data with its autoregressive and weather dependency makes it challenging to forecast using only computational intelligence methodologies such as ANN. They are not efficient for recognizing the behaviour of nonlinear time series and hence have low forecasting capability [34].

To overcome these limitations, this research has combined ANN with the LSTM model to extract the weather dependency along with the regressive aspect from the data. This study has harnessed LSTM capability of retention of information from the past and ANN capability to extract regression rules from the weather data to produce the output for obtaining more accurate prediction results for solar PV generation forecast. The prediction from each model is aggregated using the ensemble machine learning algorithm XGBoost. The advantages of using XGBoost as a meta-learner will include the ability to quantify the individual model miscalculation and data noise uncertainty that leads to higher prediction accuracy.

1.5. The novelty of the study

The contribution of this research to the literature can be summarized as:

- Developing a deep ensemble stacking model that can be used as a baseline model for solar PV generation forecast at different locations and forecasting horizons without heavy hyperparameter optimization.
- Providing a detailed comparison of individual deep learning models, bagging and stacking. As comparative application of individual deep learning and ensemble models are rarely explored in previous studies.
- Previous studies only presented the performance capability of different models without any interpretation of the predictions. In contrast, the proposed approach incorporates the SHAP analysis to understand how the model is making decisions based on the base learners.
- Finally, an evaluation is also carried out to compare the difference between the individual and aggregated forecasting of PV plants at a similar location to study the error difference.

The remainder of this paper is organized as follows: Section 2 introduces the methodological framework for the development of the stacked ensemble model. Section 3 describes the results of the modelling, and finally, section 4 concludes with general directions and considerations for future research.

2. Methodology for the solar forecast

To understand the theoretical aspects associated with the methods used please see Appendix A. Fig. 1 presents the sequence of steps required for the development and evaluation of the proposed model.

2.1. Data description

This section describes the two case studies and input features affecting the solar PV generation modelling.

2.1.1. Case study (I)

The cast study (I) data were collected with a 15 min interval from three solar farms on commercial buildings in Bunnik, Netherlands. The data collected in this study is from 424 PV panels. The Solar PV panels are TRINA type with a maximum peak capacity of 275-W peak (W_p). The panels have a flat-fix fusion south configuration and are placed on the rooftops of three buildings namely A, B, and C with a distribution of 205, 146, and 73 consecutively. The data was collected for each of the building panels separately and combined to compare the difference between the individual and aggregated forecast. Fig. 2 shows the energy generation data of all the solar PV's in kilowatt-hours (kWh). It also shows the weakly rolling average means to capture the trend of the generation. Solar PV exhibits different generation characteristics depending on the time of the year. It can be observed from the rolling mean that the generation is higher from May till September in all the test cases.

2.1.2. Case study (II)

Case study (II) was a testbed in Breda Netherlands for this project to perform experimentation. This case study consists of 65 PV panels on the rooftop of a commercial building. These PV

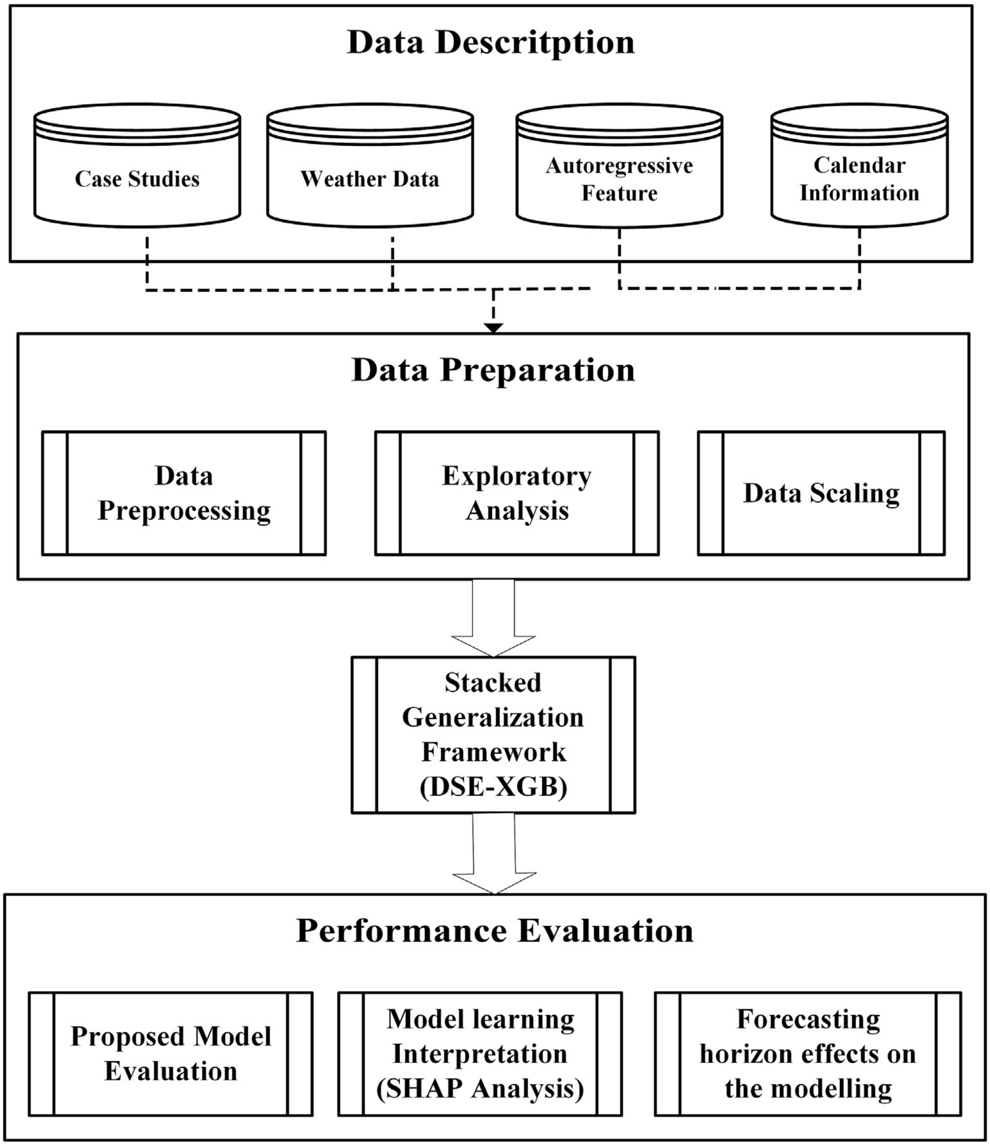


Fig. 1. Flow diagram of the model development steps.

panels were placed under an azimuth of 173°, which is an almost fully southern position. The panels are of type JAP6(SE)-60-260/3BB with a maximum output of 260 W according to Standard Test Conditions (STC) with an efficiency of 15.9%. Each PV panel is separately optimized with a DC/DC optimizer to find the Maximum PowerPoint. The data was collected with an interval of 1 h in kWh as presented in Fig. 3.

2.1.3. Weather data

The weather data obtained in this study is based on forecasted values from a weather data platform known as Solargis [35]. The platform provides long term weather data records for historical, current, and future time periods. The data from Solargis has been validated regionally and at specific sites to estimate the uncertainty in the forecasted data. The weather data was acquired for the same location as the solar panels. Solargis platform was selected for acquiring the forecasted weather data. Also, the reason for considering the forecasted weather data for training is that if the forecasted data is lower or higher than the original, the model can learn accordingly. This will avoid the over and underestimation of the forecasted generation. Hence, the prediction performance will be comparable regardless of the data used for training.

In this study, three weather parameters were selected, which were more closely related to the solar PV generation, namely global horizontal irradiance, temperature, and relative humidity. The weather variables were selected base on a literature review from a varied space of variables (Wind direction, wind speed, Dew point temperature, Air pressure) due to their strong relationships with solar PV output [36,37]. This study aimed to acquire accurate results by using fewer meteorological variables, as input for the models, to decrease the computational time of the deep learning algorithms.

2.1.4. Autoregressive features

Autoregressive features are developed using lag observation of time-series data [38]. Lags are important for time series data as the values are often correlated with a previous time step of itself. Using autoregressive features helps increase the accuracy of the models but using many variables as an input can result in overfitting during training the model [39]. Also, the addition of autoregressive features bound the forecast horizon steps of the models. If the

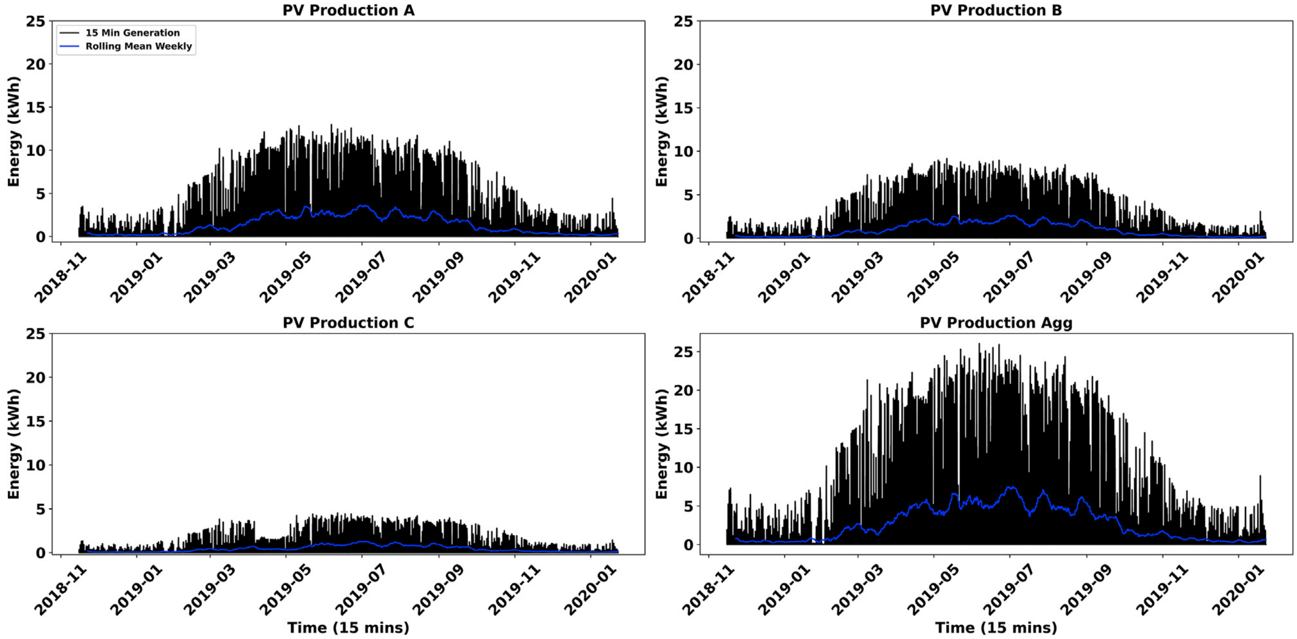


Fig. 2. PV production Case study (I)-A, B, C, and aggregated generation.

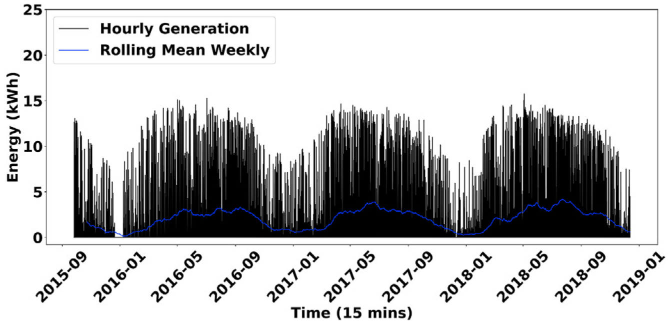


Fig. 3. PV production Case study (II).

previous time step (15 min or hour) is used as an input for the forecast then the model is limited only to a one-time step ahead forecast, subsequently, if previous day data is used then the model is only capable of a day-ahead forecast.

Considering these limitations, the modelling was done with and without the autoregressive features to show the effect on the model forecasting. Two autoregressive features were selected based on the Pearson autocorrelation function namely, point forecast (previous 15 min forecast for case study-I and previous hour for case study –II) and multi-step (previous day).

2.1.5. *Calendar information*

Solar generation data is time-series data with periodicity to time, which must be determined before the modelling. Categorical time variables were introduced considering calendar data namely, the hour of the day, day of the month, and month number to improve the result of the forecast. Overall, 8 input variables were considered given in Table 2.

2.2. *Data preparation*

This section explains in detail the data pre-processing steps taken and perform exploratory analysis on the available datasets.

Table 2
List of all input variables for both case studies.

No	Input variable	Data Type
1	Hour	Calendar information
2	Day	Calendar information
3	Month	Calendar information
4	Previous day same time	Historical PV generation (kWh)
5	Previous 15 Min (Case study (I))	Historical PV generation (kWh)
6	Previous Hour (Case study (II))	Historical PV generation (kWh)
7	Global Horizontal Irradiance (GHI)	Weather Information (W/m^2)
8	Relative Humidity (RH)	Weather Information (g/m^3)
9	Temperature (T)	Weather Information (°C)

2.2.1. *Data pre-processing*

Data pre-processing is an important step in machine learning modelling for extracting the maximum characteristic potential from a dataset. Pre-processing was done on the collected dataset for filling the missing values, checking for outliers, and scaling the features to an equivalent range. As the data in the case study (I) was collected for 15 min, observations are more likely to be similar to the previous or next observations. Therefore, linear interpolation was utilized to fill in the missing values. The data was relatively clean, so the key part of the data analysis was related to arranging the data in the right way. Case study (II) has few days of missing data at the beginning of 2016, which was interpolated with the average from the previous two days of data. Outlier detection was based on two conditions namely negative generation values and a peak capacity of the system. A visual inspection was done for looking for negative power values and output values of the peak capacity of the system. However, no outliers were identified in the dataset.

2.2.2. *Exploratory analysis*

Table 3 present the correlation matrix for the independent variables for both case studies. The output energy is positively correlated to global horizontal irradiance ‘GHI’. In solar PV forecasting, GHI is the most important variable. The ‘Temp’ parameter is also positively correlated to all the case studies; however, the correlation is weaker than the ‘GHI’. In contrast, the ‘RH’ parameter is negatively correlated to the output energy of all the case studies.

A detailed statistical description of all the indicators used for the case study (I) are presented in Table 4. The statistical analysis provides details about the mean, standard deviation, kurtosis, and skewness of the data. The kurtosis values are higher than 3 which means the data is not normally distributed and the central peak is higher and sharper [40]. The skewness values also indicate that the data is highly skewed positively since high generation events are rare in solar PV energy generation. If data have skewed dependent variables, deep learning models perform better than machine learning, unless appropriate transformations are applied to the skewed variables [41].

2.2.3. Data scaling

Machine learning models that use gradient descent optimization require data to be scaled. The data is scaled before feeding into the models for ensuring that gradient descent finds the minima smoothly and the steps for all the features are updated at a simultaneous rate. Min-Max scaler was used for scaling the dataset into a range of 0 and 1 to ensure fast convergence for the gradient learning process of ANN models. The bounded range in the Min-Max scaler will result in a smaller standard deviation by suppressing the effects of the outliers. Min-Max scaler is calculated by

$$f_{scaled} = \frac{f - f_{min}}{f_{max} - f_{min}} \tag{1}$$

Here f_{min} and f_{max} are the minimum and maximum values of the features respectively.

2.3. Stacked generalization framework (DSE-XGB)

This paper demonstrates the capability of stacked deep learning models by showing a flexible implementation considering the ensemble architecture. Stacked ensemble learning has been used in other fields (buildings energy forecasting [42,43], fault diagnosis [44], and bioinformatics [45] etc.) successfully. The main objective in stacking is finding the best combination of models for a specific application. The following tasks have been considered important for the development of the stacking ensemble model with less bias and variance: the selection of base algorithms, defining the levelling of the base models and selecting an appropriate meta learner.

- Selecting the most appropriate base learner: In every domain, an appropriate learner is selected based on some criteria, for regression tasks it is predictive accuracy. Based on the literature review; ANN and LSTM were found to be the most successful deep learning algorithms for solar PV generation forecast. Both these models are diverse with their capabilities to modify their hypothesis space for better results.
- Defining the levelling of the base models: Only one-level of stacked modelling was selected, as deep learning can be computationally heavy, and adding more layers will lead to more complex systems. Multi-level deep learning stacking may entail less benefit in accuracy relative to the computational cost.
- Selecting an appropriate meta learner: the predominant method is the estimation of the accuracy of the problem. Multiple meta-learners (linear regression, decision trees, bagging regressor, AdaBoost, SVM, XGBoost) were used for the experiments in this study for maximizing diversity. XGBoost surpass in performance for all the datasets compared to the other learning algorithms.

Table 3
Correlation of weather parameters with solar PV generation for both case studies.

Weather parameter	Case study (I)				Case study (II)
	A	B	C	Aggregated	
Temp	0.44	0.45	0.53	0.46	0.48
GHI	0.96	0.95	0.91	0.96	0.90
RH	−0.60	−0.60	−0.55	−0.52	−0.73

Different heterogeneous and homogeneous combinations of deep learning algorithms (LSTM, ANN) were tested as base models. An ensemble combination of ANN and LSTM outperforms the rest of the models in a stacked approach. A detailed description of the modelling framework is given in the following section. The framework is split into two phases for the practical implementation of this approach and explained in more detail as follows.

2.3.1. Training

Initially, the solar generation data is in the shape of *m***n* where *m* represents the number of features, output, and *n* represents the total number of observations. The data is split into training and testing in this phase. ANN and LSTM are trained on different folds of the training data using k-folds cross-validation. The k-fold cross-validation provides an unbiased estimation of the model's performance. Solar PV data has different periods throughout the year that is different from each other based on the weather. *K*-fold cross-validation training has a lower variance than a single hold-out model, which can be significant if the available data is limited. *K*-fold split the data into *k* equal-sized parts. The models are trained *k* times to predict each of the test *k*-folds. The models are validated using 5 folds cross validation. In the first iteration, the first four folds are used to train the models and the last fold is used to test the models. In the next iteration, the 2nd last fold is utilized for testing the model and the rest are used for training. This process is repeated until the predictions for all the five test folds are obtained.

The cross-validation stacking framework is exploited in the context to construct second-level data for the meta learner. Stacking uses a similar approach to cross-validation by solving two important issues: capturing diverse regions where each model performs the best and creating out-of-sample predictions. The main idea of ensemble stacking is to stack the predictions P_1, … …., P_n from the base models with weights w_1, …., w_i *(i=1, … …,n),* see equation (2)*:*

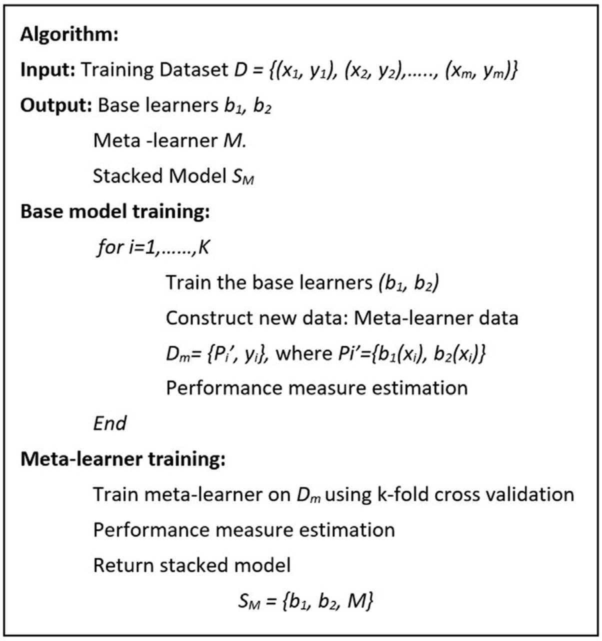
$$\boldsymbol{f_{stack}}(\boldsymbol{x}) = \sum_{\boldsymbol{i}=1}^{\boldsymbol{n}} \boldsymbol{w_i P_i} \tag{2}$$

The meta-learner is trained on predictions from the base models to learn best how to combine the predictions for the outcome. The meta-learning model is the major reason for the generalization ability of the stacking algorithm as it can distinguish where each base model is performing badly and well. XGBoost is trained as a meta-learner due to its faster computing and its ability to predict accurately with a smaller training set in a high dimensional space. The description of the whole algorithm is shown in Fig. 4.

Each of the base algorithms (b_1, b_2) is trained using 5-fold cross-validation on the training data *D* and the cross-validated predicted values are collected from each of the algorithms. The *Pi’* cross-validated predicted values from each of the algorithms are combined to form a new *m* * *p* matrix. This matrix, along with the original response vector y_i is used as the meta-learner training data. The meta-learning algorithm is trained on this new metadata D_m. The “ensemble model” S_M is based on the base models and the meta-learning model, which are then used to obtain the prediction on the test set.

Table 4
Statistical descriptions of the main indicators for case study (I).

Case study (I)							
	A	B	C	Sum	Temp	GHI	RH
Unit	*kWh*	*kWh*	*kWh*	*kWh*	°*C*	*W/m*^{2}	*g/m*^{3}
Mean	1.31	0.91	0.42	2.64	10.10	110.99	80.07
Standard deviation	2.48	1.75	0.80	4.99	6.33	188.79	13.05
Kurtosis	4.13	4.38	5.33	4.14	0.11	3.26	0.29
Skewness	2.21	2.26	2.40	2.21	0.54	1.98	−0.94



```
Algorithm:
Input: Training Dataset $D = \{(x_1, y_1), (x_2, y_2),\ldots, (x_m, y_m)\}$
Output: Base learners $b_1, b_2$
        Meta -learner $M$.
        Stacked Model $S_M$
Base model training:
        for $i=1,\ldots,K$
                Train the base learners $(b_1, b_2)$
                Construct new data: Meta-learner data
                $D_m= \{P_i', y_i\}$, where $Pi'=\{b_1(x_i), b_2(x_i)\}$
                Performance measure estimation
        End
Meta-learner training:
        Train meta-learner on $D_m$ using k-fold cross validation
        Performance measure estimation
        Return stacked model
                $S_M = \{b_1, b_2, M\}$
```

Fig. 4. DSE-XGB algorithm.

2.3.2. Testing

After the base models and the meta-learner are trained, the final predictions are obtained using the test data. The test data is unseen data that is not previously used in the training phase to get an unbiased estimation of the developed model. The trained base models predict on the test data, then the meta-learner uses the result as inputs for the final forecast. The training and testing process with the data partitioning of the stacked model is described in Fig. 5.

2.3.3. Performance measures

To evaluate the performance of the base models and ensemble scheme three commonly employed error measures in literature are used to estimate the model errors: the coefficient of determination ($R^{2)}$, root mean square error (RMSE) and mean absolute error (MAE) [26,32,46,47]. Several studies have also used Mean Absolute percentage error (MAPE), however, despite its widespread utilization, it is important to check the feasibility of the original data before selecting MAPE. Makridakis [48] stated that MAPE is asymmetric in the sense that error above the original value results in a large absolute percentage error than below the original value.

2.3.4. Development setup

The simulation was performed using the Python programming language in an open-source cross-platform integrated development environment known as Spyder [49]. For the development of ANN and LSTM, a python based open-source ANN library was utilized known as Keras [50] due to its extensible and modular focus. Keras work on top of Tensorflow [51] library and provide fast deep neural network experimentation. Tensorflow is an open-source, end-to-end machine learning platform working as an infrastructure layer for differential programming developed by google. The meta-learner used in this research was developed using the XGBoost gradient boosting library from Ref. [52]. Finally, the SHAP analysis was implemented using the game-theoretic approach from the GitHub library developed by Slundberg et al. [53].

3. Results

This section provides the results after evaluating the proposed DSE-XGB method on different case studies. It also presents the interpretation of the internal learning of the model and the effect of forecast-horizon on the modelling.

3.1. Proposed model evaluation

For a detailed comparison, the proposed algorithm was evaluated along with Bagging, ANN and LSTM. Each model was optimized using a grid search to find the optimal set of hyperparameters [54]. Hyperparameters are important for machine learning models since they control the performance of training algorithms. Table 5 refers to the hypermeters grid tested and the selected parameters for the models. After training, the test data was used for each model to validate generalization capabilities.

Figs. 6–9 illustrates the R^2, MAE, RMSE error values of the models for case study ((I) - A, B, C, and the aggregated output (Agg)). The results were obtained without using the autoregressive features and tested on 20% of the unseen data. The best models were the ones with MAE and RMSE values closer to zero and an R^2 value close to 1. The results were sorted from single deep learning algorithms to stacked ensemble learning. The figures portray the results for ANN, LSTM, bagging, and DSE-XGB. The DSE-XGB models performed better compared to other machine learning algorithms in the testing phase as discussed in section 2.3. The R-squared value results of case study (I) showed that ANN (A = 0.85, B = 0.86, C = 0.85, Agg = 0.86) and LSTM (A = 0.85, B = 0.85, C = 0.85, Agg = 0.86) perform consistently well for all the plants. The percentage difference on average for MAE and RMSE was about 1–2% for all the plants in the case study (I).

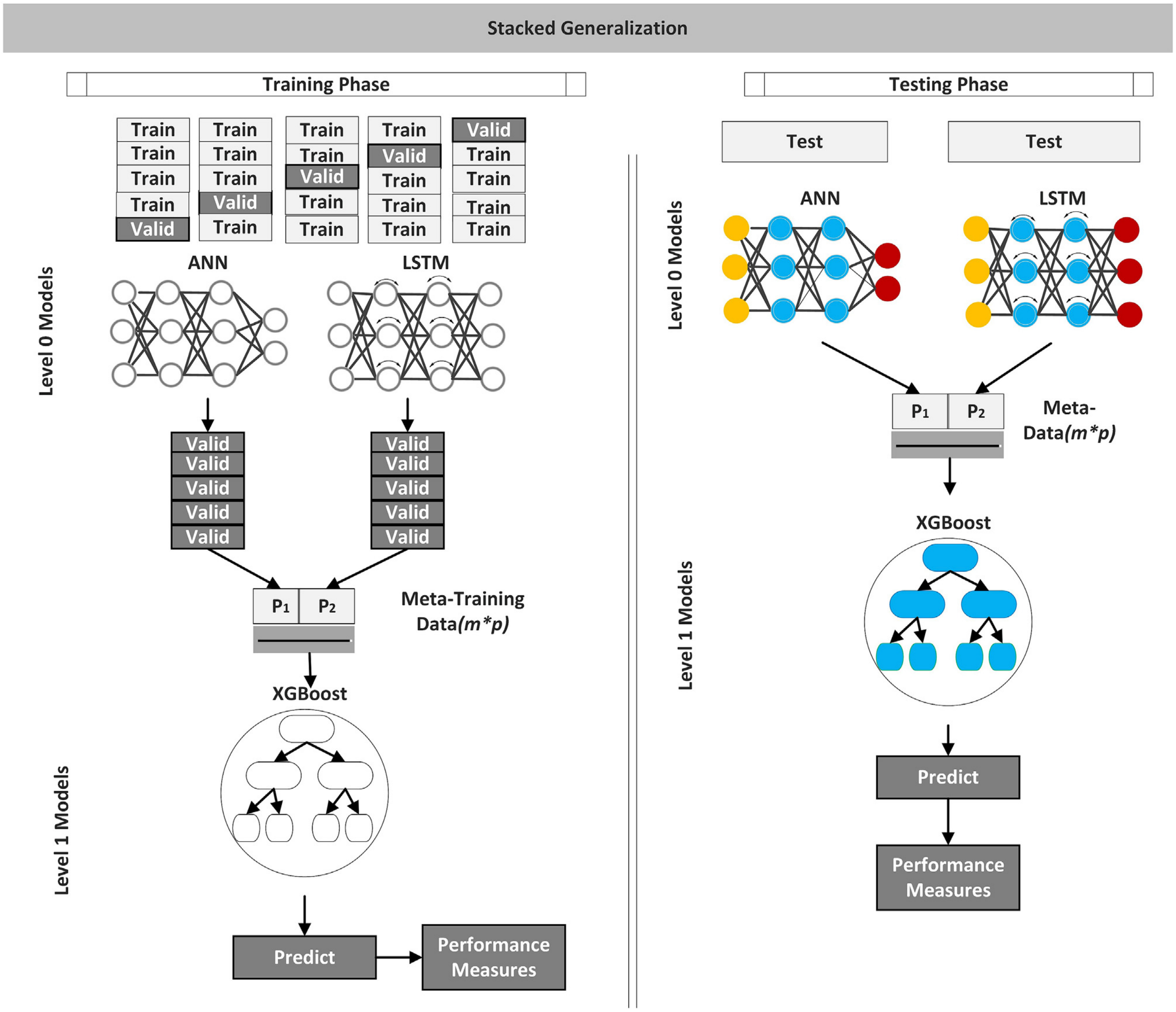


Fig. 5. DSE-XGB model with an ANN and LSTM as a base model at level 0 and XGBoost as a meta-learner at level 1.

Table 5
Selected hyperparameters, by grid search, for the trained models.

Model	Hyperparameters	Grid Search	Selected parameters
ANN	Batch size	{5,10,20,24,32,50}	10
	Epochs	{5,10,25,50,100,125,150}	100
	Activation function	{‘relu’, ‘sigmoid’, ‘tanh’, ‘linear’}	Relu
	Optimizer	{‘SGD’,‘rmsprop’, ‘adagrad’,‘adam’}	Adam
LSTM	Batch size	{5,10,20,24,32,50}	24
	Epochs	{5,10,25,50,100,125,150}	50
	Activation function	{‘relu’, ‘sigmoid’, ‘tanh’, ‘linear’}	Relu
	Optimizer	{'rmsprop','Adagrad','Adadelta','Adam','Adamax}	Adam
XGBoost	Gamma	{0.2,0.3,0.4,0.5,0.6}	0.5
	Subsample	{0.3,0.4,0.5,0.6,0.7,0.8}	0.6
	Max depth	{1,2,3,4,5,6}	2

Fig. 10 illustrate the R^2, MAE, RMSE values for case study (II). The R-squared value results of case study (II) for ANN (0.79) and LSTM (0.81) showed that there was a decline in the R-squared value of about 5–6% compared to case study (I). The percentage difference on average for MAE and RMSE was about 6–7% for case study (II). Based on the correct optimization of the hyperparameters both deep learning models performed competitively with each other for case study (I) but as the model was not optimized for case study (II) their performance declined. The models were not optimized for case study (II) to check their performance on a new dataset. Validating the hypothesis from section 1.1 that the individual model's accuracy decreases when switching from one case study to another

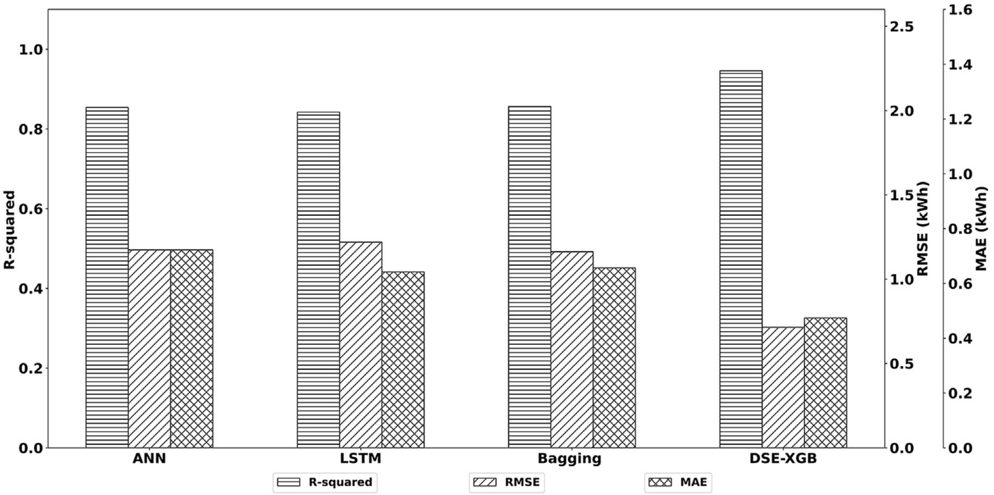


Fig. 6. Performance measures R-squared, RMSE (kWh) and MAE (kWh) for model's evaluation: Case study (I)-A.

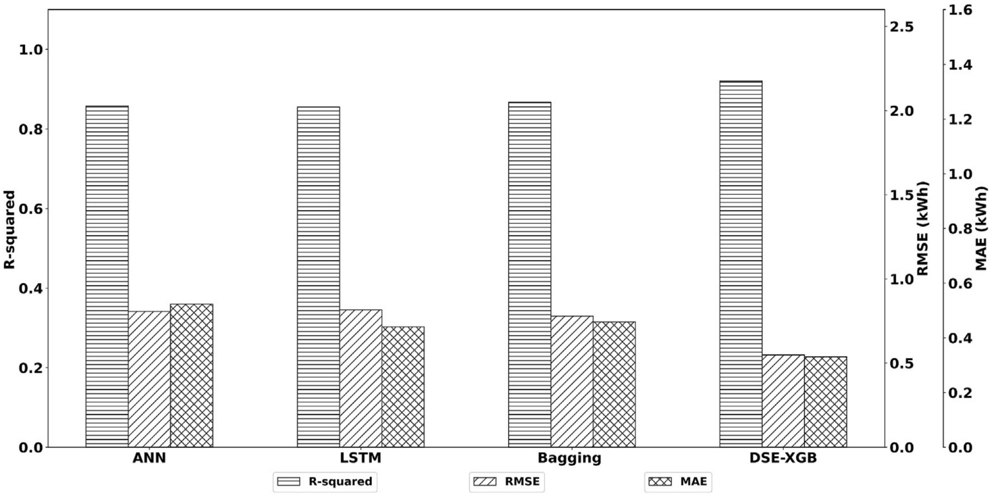


Fig. 7. Performance measures R-squared, RMSE (kWh) and MAE (kWh) for model's evaluation: Case study (I)—B.

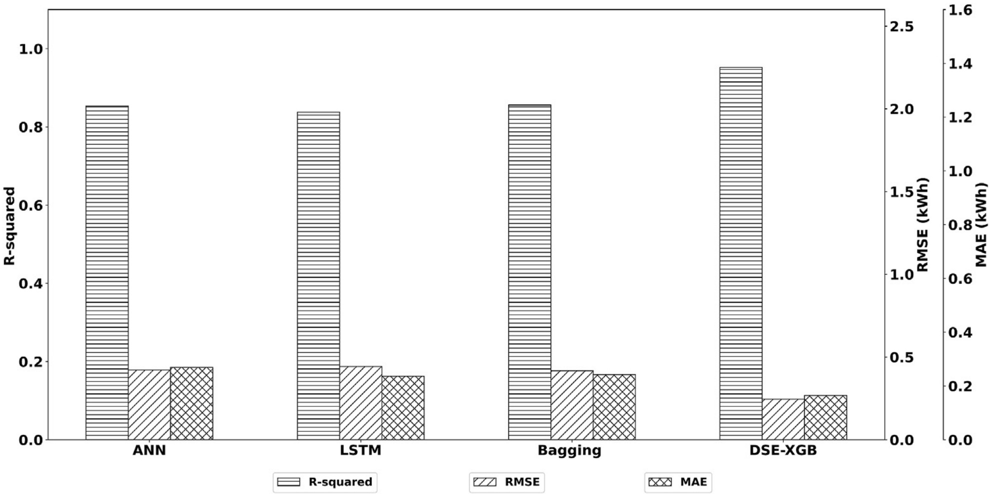


Fig. 8. Performance measures R-squared, RMSE (kWh) and MAE (kWh) for model's evaluation: Case study (I)—C.

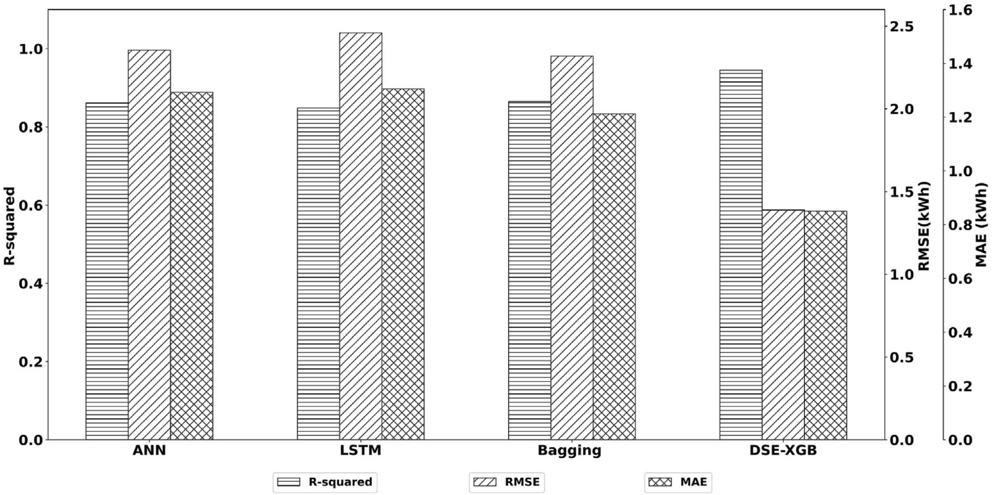


Fig. 9. Performance measures R-squared, RMSE (kWh) and MAE (kWh) for model's evaluation: Case study (I)-Agg.

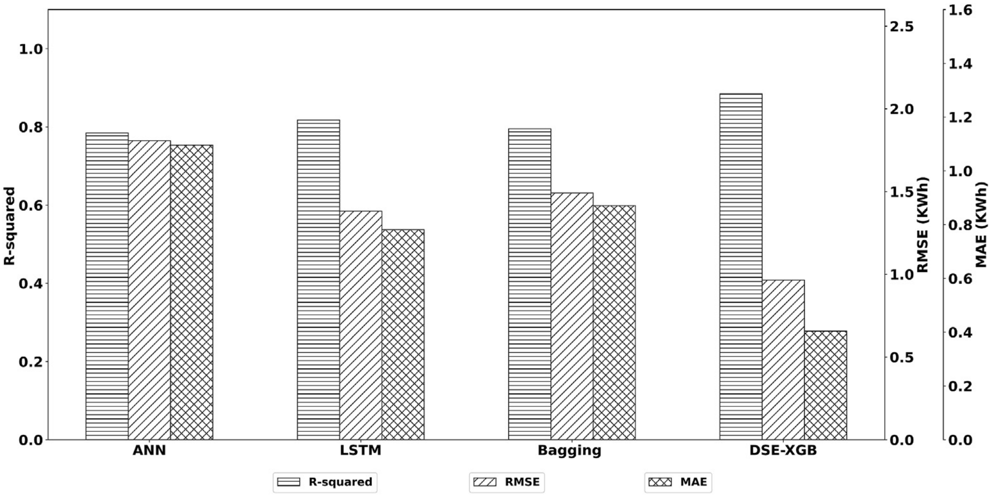


Fig. 10. Performance measures R-squared, RMSE (kWh) and MAE (kWh) for model's evaluation: Case study (II).

without proper hyperparameter optimization.

From the results, it can be observed that DSE-XGB improves the performance of the forecast for both case studies compared to the individual models and bagging. In general, the R-squared value shows that that DSE-XGB yielded better results for all the case studies. In percentage terms, the DSE-XGB had an increase in performance of prediction about 10—11%, 11—12%, 9—10% concerning ANN, LSTM, and bagging respectively for both case studies. The DSE-XGB has a stable forecast for both case studies due to its generalization capability and less reliance on the input data. The proposed model deduces the bias in the base models on both datasets and corrected that in the meta-learner. The variations in dataset and weather conditions did not affect the performance of DSE-XGB, which demonstrated the robustness of DSE-XGB for solar generation forecast.

3.1.1. *Model learning interpretation: SHAP analysis*

To identify the areas with uncertainty and determine the responsible drivers for the DSE-XGB model this study utilized the SHAP framework. The SHAP values illustrate the extent to which a given feature has changed the prediction. It allows to decompose any prediction into the sum of the effects of each feature value and explain the resulting output. The SHAP value is time consuming to compute due to its iterative nature over all possible permutations, which is factorial in the number of features. Unfortunately, it gets even worse in the case of deep learning algorithms when using a large dataset. XGBoost was used in combination with the SHAP function due to its fast convergence to generate a data-driven model [55]. To explore and open the nonlinear relationships of the black-box model and transform these relationships into interpretable rules. The resulting model enabled mapping of the learning mechanism of the meta-learner from both base model's output.

To get a deeper insight into the interactions between variables, a dependence plot was generated to see how they relate to each other in a three-dimensional space concerning SHAP values. Using predicted outputs of ANN and LSTM models as features for the DSE-XGB model, Fig. 11 provide the dependence plot for case study (I)-Agg. The x-axis is the value of the first feature (LSTM) and the y-axis represents the SHAP values attributed to each of the samples. The colour bar (z-axis) corresponds to another feature (ANN) that has a relationship with the evaluated feature. The colour code blue to red represents low to high feature values respectively.

The dependence plot demonstrates for case study (I)-Agg that, the ANN fail to predict the higher values of solar PV generation. It

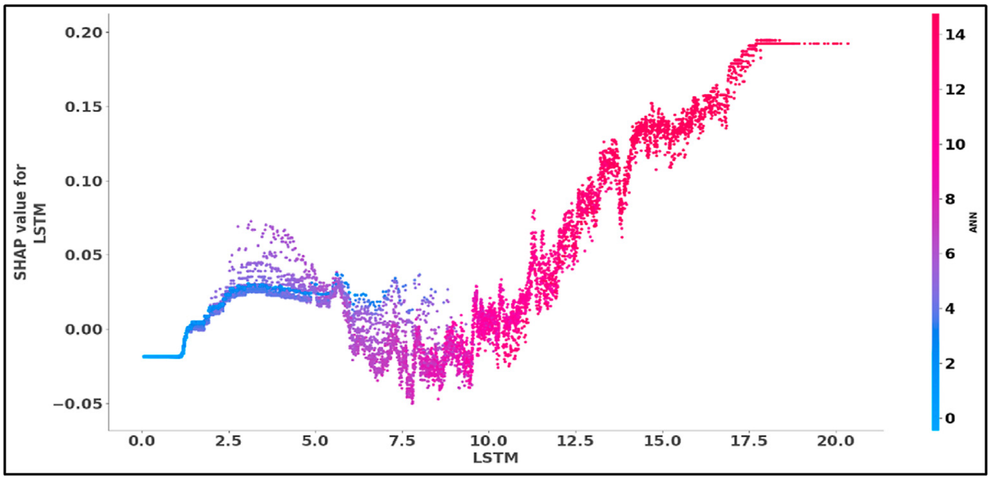


Fig. 11. SHAP feature dependence plot with interactive visualization for case study (I)-Agg.

underestimated the values on the sunny days that are around 25 kWh that is shown in Fig. 2. The lower values predicted by the ANN has a higher negative impact on the model output. It shows that the ANN was not able to predict accurately in bad weather conditions for the lower generation. On the other hand, LSTM was able to predict comparatively better in bad weather conditions and had a lower negative impact on the output of the model. Few LSTM higher values drove the model towards wrong predictions.

Fig. 12 provide the dependence plot for case study (II), the LSTM and ANN both underestimated the generation of solar PV generation for sunny days with a higher generation. The higher solar PV generation was happening rarely, and the base models could not anticipate that correctly. The meta learner was learning from the prediction of both models and managing the uncertainty in both models by correcting their mistakes.

3.1.2. *Forecasting horizon effects on the modelling*

The model was also tested using autoregressive features to compare the difference due to the forecasting horizons. The forecast was done for 15 min ahead for the case study (I), an hour ahead for case study (II) and a day ahead for both case studies.

The analysis presented in Table 6 provides a comparison for 15 min ahead and hour ahead forecast of all the models for case studies I and II respectively.

Similar to the results obtained without the autoregressive features, ANN and LSTM present comparable performance in all the error metrics for case study (I) (A, B, C, Agg), and case study (II). It can be observed from the results that short term forecasting using deep learning was stable for both case studies. There was a minor deterioration of performance when switching to another case study. The proposed DSE-XGB (comprised of base models: ANN, LSTM, and XGB) had the lowest error for case study (I)-A and I—C followed by I—B. The main comparable difference can be noticed between the case study (I)-Agg, and case study (II). There was a decline in the accuracy when switching from one location to another as highlighted in Table 6 but the overall performance was consistent compared to the base models and bagging.

In general, for 15 min ahead forecast of case study (I), the RMSE and MAE error at the aggregated level for DSE-XGB was slightly lower compared to the individual forecast. The individual RMSE and MAE error values of the case study (I)- A, B, and C amount to a sum of 0.75 kWh and 0.46 kWh, whereas the aggregated neighbourhood error was 0.74 kWh and 0.47 kWh, respectively.

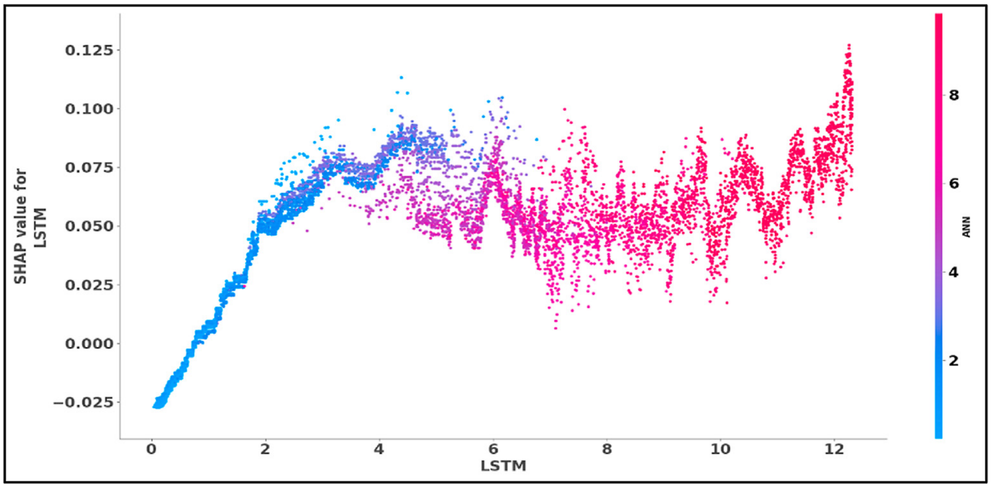


Fig. 12. SHAP feature dependence plot with interactive visualization for case study (II).

Table 6
Performance metrics R-squared, RMSE (kWh), and MAE (kWh) of all the models for a point ahead (15 min for case study (I), an hour ahead for case study (II)) forecast for both case studies.

Case Study (I)				
Error Metrics	**Model**			
	ANN	**LSTM**	**Bagging**	**DSE-XGB**
		A		
R-squared	0.92	0.91	0.92	0.99
RMSE (kWh)	0.87	0.90	0.87	0.35
MAE (kWh)	0.45	0.49	0.45	0.22
		B		
R-squared	0.92	0.92	0.92	0.98
RMSE (kWh)	0.60	0.60	0.60	0.26
MAE (kWh)	0.31	0.32	0.31	0.15
		C		
R-squared	0.92	0.92	0.92	0.99
RMSE (kWh)	0.31	0.31	0.31	0.14
MAE (kWh)	0.17	0.17	0.17	0.09
		Agg		
R-squared	**0.93**	**0.92**	**0.92**	**0.98**
RMSE (kWh)	1.73	1.83	1.74	0.74
MAE (kWh)	0.82	1.01	0.88	0.47
		Case study (II)		
R-squared	**0.90**	**0.91**	**0.89**	**0.96**
RMSE (kWh)	1.08	0.93	0.89	0.78
MAE (kWh)	1.02	0.89	0.82	0.59

Figs. 13 and 14 present the result obtained using ANN, LSTM, Bagging and DSE-XGB for one day of data compared with real values with 15 min ahead resolution for case study (I)- Agg and 1 h ahead for case study (II). For both the case studies the DSE-XGB showed consistent accuracy.

Similarly, Table 7 compares the four deep learning models for both case studies for the day ahead forecast. The ANN error results showed a noticeable variability among sub-plants in the case study (I). There was a gradual decline in performance once moving from point forecast to day-ahead forecast. The LSTM performance was stable with a minor decline in between the plants in the case study (I) and when switching to case study (II). The overall forecasting results of the individual models and bagging decreased by about 10 %—15%for day ahead forecast. However, the DSE-XGB approach outperformed the individual deep learning models and ensemble bagging. The proposed model had a variance of about 4%—5% and was holding consistently even with the change in the data at the base level. The non-reliance of deep ensemble stacking only on the input data makes it more reliable for use in solar PV generation forecast.

The error values revealed that the aggregated forecast was more accurate compared to the individual forecast of all the three plants in the case study (I). The variation in data was flattened when aggregated with other locations in a similar area, providing a more accurate prediction than the individual forecast.

Overall, this research has demonstrated that a stacked ensemble approach based on deep learning models with interpretable machine learning produces better results than any individual deep learning model and bagging on different datasets for solar PV generation forecast. It contributes to reliable modelling by reducing the risks related to the dependency on the weather parameters and uncertainty in individual modelling.

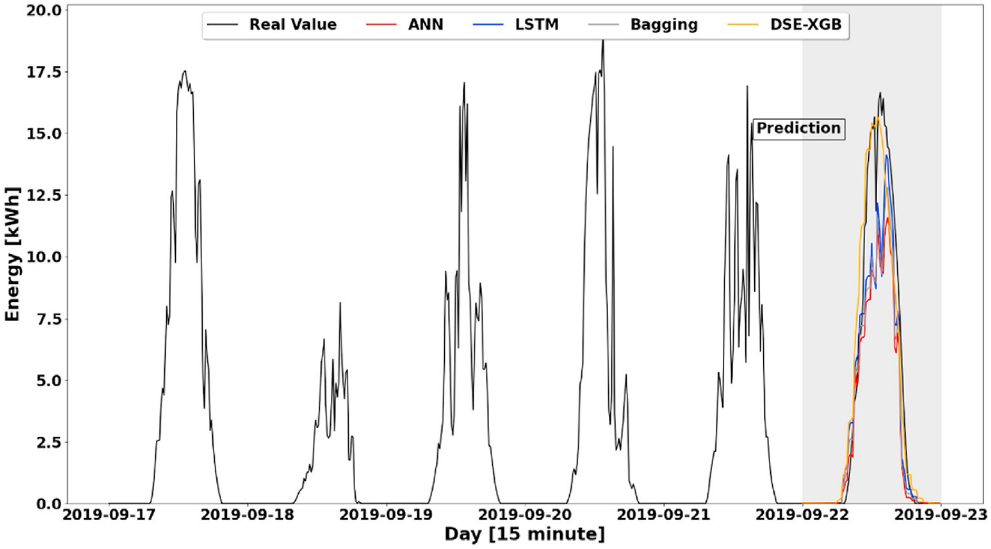


Fig. 13. Energy generation of case study (I)-Agg of one day using ANN, LSTM, Bagging, and DSE-XGB versus the true values.

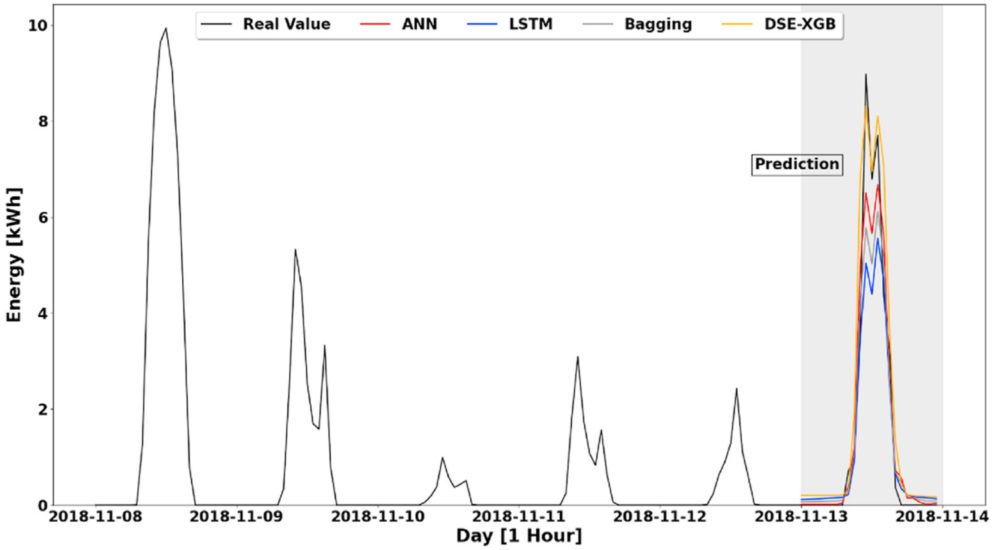


Fig. 14. Energy generation of case study (II) of one day using ANN, LSTM, Bagging, and DSE-XGB versus true values.

Table 7
Performance metrics R-squared, RMSE (kWh), and MAE (kWh) of all the models for a day-ahead forecast for both case studies.

	Case study (I)			
Error Metrics	**Model**			
	ANN	**LSTM**	**Bagging**	**DSE-XGB**
A				
R-squared	0.84	0.83	0.84	0.94
RMSE (kWh)	1.24	1.26	1.22	0.73
MAE (kWh)	0.71	0.65	0.67	0.49
B				
R-squared	0.81	0.85	0.84	0.93
RMSE (kWh)	0.94	0.84	0.88	0.47
MAE (kWh)	0.55	0.46	0.54	0.27
C				
R-squared	0.85	0.85	0.86	0.95
RMSE (kWh)	0.43	0.42	0.42	0.25
MAE (kWh)	0.25	0.22	0.23	0.16
Agg				
R-squared	0.86	0.85	0.86	0.95
RMSE (kWh)	2.40	2.44	2.37	1.30
MAE (kWh)	1.29	1.27	1.25	0.83
Case Study (II)				
R-squared	0.76	0.83	0.80	0.90
RMSE (kWh)	3.57	2.02	2.49	1.01
MAE (kWh)	1.89	1.39	1.55	0.98

4. Conclusion

In this study, an improved deep learning algorithm is proposed combining ANN, LSTM and XGBoost. The proposed DSE-XGB method outperformed the individual deep learning algorithms due to the combination of strong base learners instead of weak learners. The ANN model explicitly captured the dependency of the solar PV generation forecast while LSTM extracted the repetitive trends from the data. The prediction from each base learner was combined using a boosting approach XGBoost. The XGBoost meta-learner made sense of the base models' outputs to generalize on the testing data. The meta learner was trained only after the ensemble had completed training for the base models. The XGBoost as a meta-learner builds tree sequentially such that each subsequent tree aims to reduce the errors of the previous trees. Each tree learns from its predecessors and updates the residual errors. Each of the base learners provided vital information for prediction and enabled the XGBoost to manage the uncertainty by effectively combining the output from these strong learners. This study has integrated interpretable machine learning with the modelling to understand how the meta-learner learns from the base model predictions.

The meta learner brings down both the variance and the bias by correcting the errors of the previous models and selecting their strong areas for the final prediction. Data uncertainty or irreducible uncertainty can always degrade the performance of the individual deep learning model. The introduction of more data might not reduce uncertainty and can slow the convergence of the models. Increasing measurement precision or designing models that do not rely completely on the input features can manage the uncertainty in the forecast. The proposed DSE-XGB model doesn't completely rely on the input features and can handle uncertainty in the forecast from the individual models. The proposed DSE-XGB model counters the variance of the individual models by generating predictions that are less sensitive to the specifics of the training data, the optimization of the individual models, the providence of the training run, and the choice of the training scheme.

The successful application of the proposed stacking ensemble algorithm can be used by generalizing problems from other fields such as medicine, control engineering, and financial markets. Therefore, DSE approaches should be studied in more detail to be applied on a larger scale to varied datasets for obtaining consistent results.

To conclude, some recommendations for solar forecasting research and future work are: (i) The ANN and LSTM were selected based on the literature review, however, there is still room for improvement, mainly because deep learning models have many variations (DBP, RBM, AE, CNN, etc.) that can be used to select more accurate models. (ii) Another aspect that should be considered in the selection of the base models could be the trade-off between prediction accuracy gain and increase in computation time. (iii) It would also be interesting to evaluate the proposed model in real time and evaluate its performance and practical applicability with building energy management systems.

Author statement

Waqas Khan – Conceptualization, Methodology, Development, Validation, Writing – original draft writing, Visualization. Shalika Walker – Validation, Writing, Supervision. Wim Zeiler – Supervision, Funding acquisition

Declaration of competing interest

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

Acknowledgements

This study was funded by the Dutch Research Council (NWO). NWP weather data used in this study was provided by Solargis. The other data used in the study is supplied by two construction companies from Breda and Bunnik, Netherlands and is only available in the form of tables and graphs presented in the study because of the restrictions on the use by third parties.

Appendix A

This section describes the theoretical aspects associated with the used methods.

A.1 Ensemble learning

The deployment of different combination schemes or different base learner implementation leads to different ensemble methods. In this aspect, this section aims to illustrate the key aspects and concepts of those methodologies, which are required for the understanding of this paper.

A.1.1 Bagging

Bagging or bootstrapping is one of the simplest methods in ensemble learning proposed by Breiman [56]. In bagging several subsets of data are chosen randomly from the training set with replacement. Each subset of the data is selected by sampling from the total M data sample, choosing M items at random uniformly with replacement. Then either a classification or regression algorithm is trained on each subset. Finally, in the case of classification, the most voted output is accepted and in the case of regression, an average is taken of all the individual outcomes by the models.

Bagging works best with high variance models, which are unstable such as KNN and DT as they produce different generalization behaviour with small changes to the data. It doesn't work well with simple models such as linear regression, the results generated are almost identical from the sampling [57,58]. The advantage of using bagging is that it creates its variance by sampling the data and testing multiple hypotheses to solve the problem of overfitting. In general, the main objective of bagging is to increase the performance of models by reducing their variance [59].

A.1.2 Stacking

Stacked generalization or stacking is another ensemble learning technique introduced by Wolpert [60] that has been widely used in multiple fields since its inception. In stacking the outcome of different models (logistic regression, SVM, ANN, etc.) are combined to train a new meta-learner for the outcome. The basic principle of stacking is based on two levels of algorithms. The first level consists of different algorithms known as base learners and the second level consists of a stacking algorithm known as meta-learner. The first-level learners are often diverse different learning algorithms; however, stacked ensembles can also be generated from the same learning algorithms [61]. The base learners are trained on the original dataset to predict the outcome. The prediction of each base learner is collected to create a new dataset. The new dataset consists of the prediction by the base learners. The second level meta-learner use this dataset to provide the final prediction. The aim of the meta-learner model to correct the output prediction, thereby correcting any errors made by the base models. Staking can have several levels, where the prediction from one level acts as an input for the next. In ensemble learning, stacking is the best state of the art method. It can decrease both variance and bias efficiently by avoiding overfitting.

The following section gives a brief introduction of the base models and meta-learners used in the stacking algorithm in this research.

A.1.3 Reference models

In this section, the general structure and characteristics of reference models are described.

A.1.3.1 ANN. ANN's are built from adaptive processing elements known as neurons that are inspired by the abilities of the human brain. These processing elements can modify their internal structure concerning a function objective. The ANN's are formed from many interconnected layers of neurons called 'Multilayer Perceptron'. There has been a huge increase in interest in ANN's during the last decade. Several types of ANN's have been proposed since its development, but they all have three things in common: the individual neuron, connections between the neurons, and the learning algorithm [62]. Each network type limits the kind of connections that are possible. ANN's have been employed successfully in a range of functional tasks ranging from robotics, fault detection, power systems, process control, signal processing, and pattern recognition [63].

ANNs are utilized in predictive modelling applications due to their learning and universal mapping capability. They can develop a generalized solution for problems other than the ones learned during the training and generate reliable solutions, even if the training data contain errors.

A.1.3.2 Long short-term memory. To introduced the time concept to the traditional ANN's and to make them more adaptive to time horizon dependency, a new type of neural network was introduced by John Hopfield in 1982 known as recurrent neural networks (RNN) [64]. An RNN is different from a feedforward ANN in the sense that it has at least one feedback loop. The performance and learning capability of a feed-forward ANN increase profoundly with the addition of a feedback loop. The effective behaviour of RNN's is due to the presence of feedback connections, which enables the processing of time-dependent patterns in the sense that the output at a given time depends on past data values. However, RNN's have a drawback of vanishing gradient, which obstructs the learning of long data sequences. In RNN's the gradients carry information that is required for parameter updates and when the gradient becomes too smaller, no real learning is done as the parameter's updates become insignificant. To solve the problem of vanishing gradient Hochreiter et el proposed LSTM networks that are an enhanced version of RNN and designed to easily capture long-term dependency in data sequences [65]. The hidden state in a regular RNN is influenced by the nearest local activation known as short term memory. Whereas the network weights are influenced by the computations over the entire long data sequence known as long term memory. LSTM was designed to preserve information over long distances and has an activation state that can act as weights. LSTM has been successfully applied in a lot of fields (specifically in time series problems) such as machine translations, speech recognition, stock prediction, and language modelling [66,67].

A.1.3.3 XGBoost. The extreme gradient boosting is a supervised machine learning algorithm based on boosted trees proposed by Chen and Guestrin in 2016 [68]. XGBoost is an improved and scalable version of the GB algorithm that iteratively merges weak base learner models into a stronger learner. In XGBoost the input data is fitted to the first learner. To boost the learning ability of the first learner, a second model is fitted to the residual of that previous model. This residual accumulation process is repeated until the predetermined conditions are met. The result is calculated by adding the outcomes from all the learners. By incorporating a regularization term to the objective function, it also avoids overfitting. The learning process is faster in XGBoost compared to GB due to its system optimization, parallel, and distributed computing [69]. GB employs a stopping criterion for tree splitting criteria, which depends on a negative loss criterion whereas the XGBoost algorithm uses a depth-first approach. The XGBoost prunes the tree in a backward direction by using the max depth parameter. In the XGBoost algorithm, the process of sequential tree building is done by a parallelized implementation. The inner and outer loops of XGBoost are interchangeable. The inner loop calculates the feature of a tree while the outer loop lists the leaf nodes. This switching system enhances the algorithm's efficiency. XGBoost has been implemented in many fields and Kaggle competition due to its minimal requirement and high performance for problem solving [70].

A.1.3.4 SHapley Additive exPlanations (SHAP). Understanding why machine learning models make certain predictions can be crucial for studying the uncertainty in solar generation forecasts. SHAP is a method proposed by Lundberg and Lee based on the game theory approach *Shapely* to explain the individual predictions of machine learning models [55]. The framework unified the previously proposed methods (Shapley sampling values, LIME, DeepLIFT, Layer-Wise Relevance Propagation, Shapley regression values, and tree interpreter) for interpreting machine learning models [55]. The SHAP framework utilized concepts from cooperative game theory, by assigning each attribute an importance score based on its impact on the model forecast. To explain complex models, SHAP uses a linear additive feature attribute technique as a simpler explanation model.

SHAP builds a simplified descriptive model σ to describe the prediction f using $x'_i \in \{0, 1\}^N$, where i is the instance and N is the number of features. The local explanatory model is given by:

$$\sigma\,(x') = \varphi_0 + \sum_{i=1}^{N} \varphi_i\, x'_i \tag{1}$$

The local feature effect (SHAP values) of feature i is φ_i and is calculated for all possible input samples. x'_i is the simplified input vector that indicates if a particular feature is present or not during the estimation, and φ_0 is associated with the model prediction when all the attributes are not considered in the estimation.

A surprising attribute of the SHAP methods is the presence of a single unique solution in this class with three desirable properties: local accuracy, missingness, and consistency. SHAP combine these conditional properties with the classic Shapley values to attribute φ_i values to each feature. Only one possible model satisfies all three properties.

$$\varphi_i(f,x) = \sum_{z\subseteq x'} \frac{|z'|\,!\,(N-|z'|-1)!}{N!}\,[f_x(z') - f_x\,(z'\,/\,i)] \tag{2}$$

where $|z'\,|$ is the number of non-zero entries in z', and $z' \subseteq x'$ represents all z' vectors where the non-zero entries are a subset of the non-zero entries in x' .

References

[1] https://www.irena.org/publications/2020/Jun/Renewable-Power-Costs-in-2019; 2020.

[2] IRENA. Future of Solar Photovoltaic: deployment, investment, technology, grid integration and socio-economic aspects (A Global Energy Transformation: paper). 2019.

[3] Walker S, Bergkamp V, Yang D, van Goch TAJ, Katic K, Zeiler W. Improving energy self-sufficiency of a renovated residential neighborhood with heat pumps by analyzing smart meter data. Energy 2021;229. https://doi.org/10.1016/j.energy.2021.120711.

[4] https://cleanenergysolutions.org/resources/variable-renewable-energy-forecasting-integration-electricity-grids-markets-best-practice; 2015.

[5] Shi J, Lee WJ, Liu Y, Yang Y, Wang P. Forecasting power output of photovoltaic systems based on weather classification and support vector machines. IEEE Trans Ind Appl 2012;48:1064–9. https://doi.org/10.1109/TIA.2012.2190816.

[6] Lai JP, Chang YM, Chen CH, Pai PF. A survey of machine learning models in renewable energy predictions. Appl Sci 2020;10(17). https://doi.org/10.3390/app10175975.

[7] Bogner K, Pappenberger F, Zappa M. Machine learning techniques for predicting the energy consumption/production and its uncertainties driven by meteorological observations and forecasts. Sustainability 2019;11(12). https://doi.org/10.3390/su11123328.

[8] Ahmad MW, Mourshed M, Rezgui Y. Tree-based ensemble methods for predicting PV power generation and their comparison with support vector regression. Energy 2018;164:465–74. https://doi.org/10.1016/j.energy.2018.08.207.

[9] Thevenard D, Pelland S. Estimating the uncertainty in long-term photovoltaic yield predictions. Sol Energy 2013;91:432–45. https://doi.org/10.1016/j.solener.2011.05.006.

[10] Ahmed R, Sreeram V, Mishra Y, Arif MD. A review and evaluation of the state-of-the-art in PV solar power forecasting: techniques and optimization. Renew Sustain Energy Rev 2020;124. https://doi.org/10.1016/j.rser.2020.109792.

[11] Ahmad T, Zhang H, Yan B. A review on renewable energy and electricity requirement forecasting models for smart grid and buildings. Sustain Cities Soc 2020;55. https://doi.org/10.1016/j.scs.2020.102052.

[12] Wang H, Liu Y, Zhou B, Li C, Cao G, Voropai N, et al. Taxonomy research of artificial intelligence for deterministic solar power forecasting. Energy Convers Manag 2020;214(3). https://doi.org/10.1016/j.enconman.2020.112909.

[13] Wang J, Li P, Ran R, Che Y, Zhou Y. A short-term photovoltaic power prediction model based on the Gradient Boost Decision Tree. Appl Sci 2018;8(5). https://doi.org/10.3390/app8050689.

[14] Persson C, Bacher P, Shiga T, Madsen H. Multi-site solar power forecasting using gradient boosted regression trees. Sol Energy 2017;150:423–36. https://doi.org/10.1016/j.solener.2017.04.066.

[15] Leva S, Dolara A, Grimaccia F, Mussetta M, Ogliari E. Analysis and validation of 24 hours ahead neural network forecasting of photovoltaic output power. Math Comput Simulat 2017;131:88–100. https://doi.org/10.1016/j.matcom.2015.05.010.

[16] Munawar U, Wang Z. A framework of using machine learning approaches for short-term solar power forecasting. J Electr Eng Technol 2020;15:561–9. https://doi.org/10.1007/s42835-020-00346-4.

[17] Zhang J, Florita A, Hodge BM, Lu S, Hamann HF, Banunarayanan V, et al. A suite of metrics for assessing the performance of solar power forecasting. Sol Energy 2015;111:157–75. https://doi.org/10.1016/j.solener.2014.10.016.

[18] Andrade JR, Bessa RJ. Improving renewable energy forecasting with a grid of numerical weather predictions. IEEE Trans Sustain Energy 2017;8(5):1571–80. https://doi.org/10.1109/TSTE.2017.2694340.

[19] Li P, Zhou K, Lu X, Yang S. A hybrid deep learning model for short-term PV power forecasting. Appl Energy 2020;259. https://doi.org/10.1016/j.apenergy.2019.114216.

[20] Dairi A, Harrou F, Sun Y, Khadraoui S. Short-term forecasting of photovoltaic solar power production using variational auto-encoder driven deep learning approach. Appl Sci 2020;10(23):1–20. https://doi.org/10.3390/app10238400.

[21] Nitisanon S, Hoonchareon N. Solar power forecast with weather classification using self-organized map. IEEE Power Energy Soc. Gen. Meet. 2018;2018:1–5. https://doi.org/10.1109/PESGM.2017.8274548.

[22] De V, Teo TT, Woo WL, Logenthiran T. Photovoltaic power forecasting using LSTM on limited dataset. In: Int. Conf. Innov. Smart grid technol. ISGT asia

2018; 2018. https://doi.org/10.1109/ISGT-Asia.2018.8467934.

[23] Abuella M, Chowdhury B. Solar power forecasting using artificial neural networks. In: 2015 north Am. Power symp. NAPS 2015; 2015. https://doi.org/10.1109/NAPS.2015.7335176.

[24] Chen C, Duan S, Cai T, Liu B. Online 24-h solar power forecasting based on weather type classification using artificial neural network. Sol Energy 2011;85(11):2856–70. https://doi.org/10.1016/j.solener.2011.08.027.

[25] Almonacid F, Pérez-Higueras PJ, Fernández EF, Hontoria L. A methodology based on dynamic artificial neural network for short-term forecasting of the power output of a PV generator. Energy Convers Manag 2014;85:389–98. https://doi.org/10.1016/j.enconman.2014.05.090.

[26] Vaz AGR, Elsinga B, van Sark WGJHM, Brito MC. An artificial neural network to assess the impact of neighbouring photovoltaic systems in power forecasting in Utrecht, The Netherlands. Renew Energy 2016;85:631–41. https://doi.org/10.1016/j.renene.2015.06.061.

[27] Lin Y, Koprinska I, Rana M, Troncoso A. Pattern sequence neural network for solar power forecasting. Commun. Comput. Inf. Sci 2019;1143:727–37. https://doi.org/10.1007/978-3-030-36802-9_77.

[28] Wang K, Qi X, Liu H. A comparison of day-ahead photovoltaic power forecasting models based on deep learning neural network. Appl Energy 2019;251(C):1. https://doi.org/10.1016/j.apenergy.2019.113315.

[29] Zhou ZH. Ensemble methods: Foundations and algorithms. 2012. https://doi.org/10.1201/b12207.

[30] Pierro M, Bucci F, De Felice M, Maggioni E, Moser D, Perotto A, et al. Multi-Model Ensemble for day ahead prediction of photovoltaic power generation. Sol Energy 2016;134:132–46. https://doi.org/10.1016/j.solener.2016.04.040.

[31] Kumari P, Toshniwal D. Extreme gradient boosting and deep neural network based ensemble learning approach to forecast hourly solar irradiance. J Clean Prod 2021;279. https://doi.org/10.1016/j.jclepro.2020.123285.

[32] Gensler A, Henze J, Sick B, Raabe N. Deep Learning for solar power forecasting - an approach using AutoEncoder and LSTM Neural Networks. In: 2016 IEEE int. Conf. Syst. Man, cybern. SMC 2016-conf. Proc.; 2017. https://doi.org/10.1109/SMC.2016.7844673.

[33] Qing X, Niu Y. Hourly day-ahead solar irradiance prediction using weather forecasts by LSTM. Energy 2018;148:461–8. https://doi.org/10.1016/j.energy.2018.01.177.

[34] Tealab A, Hefny H, Badr A. Forecasting of nonlinear time series using ANN. Futur Comput Informatics J 2017;2(1):39–47. https://doi.org/10.1016/j.fcij.2017.05.001.

[35] SolarGIS. SolarGIS. Imaps. Free Sol Radiat Inf - GHI. 2011.

[36] Kazem HA, Chaichan MT. Effect of humidity on photovoltaic performance based on experimental study. Int J Appl Eng Res 2015;10(23):43572–7.

[37] AlSkaif T, Dev S, Visser L, Hossari M, van Sark W. A systematic analysis of meteorological variables for PV output power estimation. Renew Energy 2020;153:12–22. https://doi.org/10.1016/j.renene.2020.01.150.

[38] Autoregressive Moving Average Models. https://doi.org/10.1002/9781118032466.ch3; 2011.

[39] Moon J, Jung S, Rew J, Rho S, Hwang E. Combination of short-term load forecasting models based on a stacking ensemble approach. Energy Build 2020;216(1). https://doi.org/10.1016/j.enbuild.2020.109921.

[40] Joanes DN, Gill CA. Comparing measures of sample skewness and kurtosis. J R Stat Soc - Ser D Statistician 1998;47(1):183–9. https://doi.org/10.1111/1467-9884.00122.

[41] Subbanarasimha PN, Arinze B, Anandarajan M. Predictive accuracy of artificial neural networks and multiple regression in the case of skewed data: exploration of some issues. Expert Syst Appl 2000;19:117–23. https://doi.org/10.1016/S0957-4174(00)00026-9.

[42] Rahman S, Rabiul Alam MG, Mahbubur Rahman M. Deep learning based ensemble method for household energy demand forecasting of smart home. In: 2019 22nd int. Conf. Comput. Inf. Technol. ICCIT 2019; 2019. https://doi.org/10.1109/ICCIT48885.2019.9038565.

[43] Khan W, Walker S, Katic K, Zeiler W. A novel framework for autoregressive features selection and stacked ensemble learning for aggregated electricity demand prediction of neighborhoods. In: SEST 2020–3rd int. Conf. Smart energy syst. Technol; 2020. https://doi.org/10.1109/SEST48500.2020.9203507.

[44] Li J, Lin M. Ensemble learning with diversified base models for fault diagnosis in nuclear power plants. Ann Nucl Energy 2021;158. https://doi.org/10.1016/j.anucene.2021.108265.

[45] Ma Z, Wang P, Gao Z, Wang R, Khalighi K. Ensemble of machine learning algorithms using the stacked generalization approach to estimate the warfarin dose. PLoS One 2018;13. https://doi.org/10.1371/journal.pone.0205872.

[46] Wang D, Luo H, Grunder O, Lin Y, Guo H. Multi-step ahead electricity price forecasting using a hybrid model based on two-layer decomposition technique and BP neural network optimized by firefly algorithm. Appl Energy 2017;190:390–407. https://doi.org/10.1016/j.apenergy.2016.12.134.

[47] Bouzerdoum M, Mellit A. Massi Pavan A. A hybrid model (SARIMA-SVM) for short-term power forecasting of a small-scale grid-connected photovoltaic plant. Sol Energy 2013;98(C):226–35. https://doi.org/10.1016/j.solener.2013.10.002.

[48] Makridakis S. Accuracy measures: theoretical and practical concerns. Int J Forecast 1993;9(4):527–9. https://doi.org/10.1016/0169-2070(93)90079–3.

[49] Spyder. SPYDER IDE. Spyder Proj. 2018.

[50] Chollet F. Keras: the Python deep learning library. KerasIo 2015.

[51] Abadi M, Barham P, Chen J, Chen Z, Davis A, Dean J, et al. TensorFlow: a system for large-scale machine learning. In: Proc. 12th USENIX symp. Oper. Syst. Des. Implementation, OSDI 2016; 2016.

[52] Chen T, Cho H. GitHub - dmlc/xgboost: Scalable, Portable and Distributed Gradient Boosting (GBDT, GBRT or GBM) Library, for Python, R, Java, Scala, C++ and more. Runs on single machine, Hadoop, Spark, Dask, Flink and DataFlow n.d. https://github.com/dmlc/xgboost (accessed September 28, 2021).

[53] slundberg scott. GitHub - slundberg/shap: a game theoretic approach to explain the output of any machine learning model. 2017. https://ProceedingsNeuripsCc/Paper/2017/Hash/8a20a8621978632d76c43dfd28b67767-AbstractHtml. https://github.com/slundberg/shap. [Accessed 28 September 2021].

[54] Hutter F. Automated machine learning. 2019.

[55] Lundberg SM, Lee SI. A unified approach to interpreting model predictions. Adv Neural Inf Process Syst 2017;30:4768–77.

[56] Breiman L. Bagging predictors. Mach learn. 1996. https://doi.org/10.1007/bf00058655.

[57] Brown G. Ensemble learning. In: Sammut C, Webb GI, editors. Encycl. Mach. Learn. Boston, MA: Springer; 2010. p. 312–20. https://doi.org/10.1007/978-0-387-30164-8_252. US.

[58] Altman N, Krzywinski M. Ensemble methods: bagging and random forests. Nat Methods 2017;14:933–4. https://doi.org/10.1038/nmeth.4438.

[59] Assouline D, Mohajeri N, Scartezzini JL. Large-scale rooftop solar photovoltaic technical potential estimation using Random Forests. Appl Energy 2018;217:189–211. https://doi.org/10.1016/j.apenergy.2018.02.118.

[60] Wolpert D. Stacked generalization (stacking). Neural Network 1992;5(2):241–59. https://doi.org/10.1016/S0893-6080(05)80023-1.

[61] Breiman L. Stacked regressions. Mach learn. 1996. https://doi.org/10.1007/bf00117832.

[62] Cook TR. Neural networks. Adv. Stud. Theor. Appl. Econom 2020;52:161–89. https://doi.org/10.1007/978-3-030-31150-6_6.

[63] Kalogirou SA. Applications of artificial neural-networks for energy systems. Appl Energy 2000;67:17–35. https://doi.org/10.1016/S0306-2619(00)00005–2.

[64] Hopfield JJ. Neural networks and physical systems with emergent collective computational abilities. Feynman Comput 2018;79(8):2554–8. https://doi.org/10.1201/9780429500459.

[65] Hochreiter S, Schmidhuber J. Long short-term memory. Neural Comput 1997;9(8):1735–80. https://doi.org/10.1162/neco.1997.9.8.1735.

[66] Cho K, Van Merriënboer B, Gulcehre C, Bahdanau D, Bougares F, Schwenk H, et al. Learning phrase representations using RNN encoder-decoder for statistical machine translation. In: EMNLP 2014-2014 conf. Empir. Methods nat. Lang. Process. Proc. Conf.; 2014. https://doi.org/10.3115/v1/d14-1179.

[67] Nelson DMQ, Pereira ACM, De Oliveira RA. Stock market's price movement prediction with LSTM neural networks. In: Proc. Int. Jt. Conf. Neural networks; 2017. https://doi.org/10.1109/IJCNN.2017.7966019.

[68] Chen T, Guestrin C. XGBoost: a scalable tree boosting system. Proc. ACM SIGKDD Int. Conf. Knowl. Discov. Data Min 2016;22:785–94. https://doi.org/10.1145/2939672.2939785.

[69] Nielsen D. Tree boosting with XGBoost why Does XGBoost Win "every" machine learning competition? Tree boost with XGBoost - why Does XGBoost Win “every” Mach learn Compet. 2016.

[70] Pan B. Application of XGBoost algorithm in hourly PM2.5 concentration prediction. IOP Conf Ser Earth Environ Sci 2018;113(1). https://doi.org/10.1088/1755-1315/113/1/012127.