



Article

A Novel Feature Representation for Prediction of Global Horizontal Irradiance Using a Bidirectional Model

Sourav Malakar^{1,*}, Saptarsi Goswami^{2,†}, Bhaswati Ganguli³, Amlan Chakrabarti¹, Sugata Sen Roy³, K. Boopathi⁴ and A. G. Rangaraj⁴

¹ A.K. Choudhury School of Information Technology, University of Calcutta, Kolkata 700106, India; achakra12@yahoo.com

² Bangabasi Morning College, University of Calcutta, Kolkata 700009, India; saptarsi007@gmail.com

³ Department of Statistics, University of Calcutta, Kolkata 700019, India; bgstat@gmail.com (B.G.); sugatasr@gmail.com (S.S.R.)

⁴ National Institute of Wind Energy (NIWE), Under the Ministry of New and Renewable Energy, Chennai 600100, India; boopathi.niwe@nic.in (K.B.); rangaraj.niwe@nic.in (A.G.R.)

* Correspondence: sourav.xaviers@gmail.com

† These authors contributed equally to this work.

Abstract: Complex weather conditions—in particular clouds—leads to uncertainty in photovoltaic (PV) systems, which makes solar energy prediction very difficult. Currently, in the renewable energy domain, deep-learning-based sequence models have reported better results compared to state-of-the-art machine-learning models. There are quite a few choices of deep-learning architectures, among which Bidirectional Gated Recurrent Unit (BGRU) has apparently not been used earlier in the solar energy domain. In this paper, BGRU was used with a new augmented and bidirectional feature representation. The used BGRU network is more generalized as it can handle unequal lengths of forward and backward context. The proposed model produced 59.21%, 37.47%, and 76.80% better prediction accuracy compared to traditional sequence-based, bidirectional models, and some of the established states-of-the-art models. The testbed considered for evaluation of the model is far more comprehensive and reliable considering the variability in the climatic zones and seasons, as compared to some of the recent studies in India.

Keywords: GHI forecasting; time series; bidirectional features; bidirectional GRU



Citation: Malakar, S.; Goswami, S.; Ganguli, B.; Chakrabarti, A.; Roy, S.S.; Boopathi, K.; Rangaraj, A.G. A Novel Feature Representation for Prediction of Global Horizontal Irradiance Using a Bidirectional Model. *Mach. Learn. Knowl. Extr.* **2021**, *3*, 946–965.

<https://doi.org/10.3390/make3040047>

Academic Editor: Salim Lahmiri

Received: 21 October 2021

Accepted: 23 November 2021

Published: 25 November 2021

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Solar energy is one of the important components of the alternative sources of energy [1]. India is ranked third after China and the United States of America (USA) in terms of solar energy development [2]. Precise prediction of solar energy is very important for several applications, such as electricity grid management, the trading of solar energy, etc. [3,4]. The Global Horizontal Irradiance (GHI) is often taken as a proxy for solar energy generation and used for the prediction task [5–12]. A considerable amount of uncertainty is present in solar energy due to its strong dependence on atmospheric conditions, which makes the prediction task challenging [13–15].

In [10], the authors categorized solar forecasting horizons into short-term, medium-term, and long-term forecasting. For one to few hours ahead of solar forecasting, i.e., for short-term forecasting, currently, machine-learning models are the state-of-the-art models [14]. In [16], the authors stated that short-term solar forecasting is essential for balancing demand and supply and decreasing the storage requirement, unit commitment, etc. From the literature study, we found that statistical and Artificial Neural Network (ANN)-based models were effective in intra-hour (short-term) or intra-day (medium-term) solar forecasting.

Many of the current works suggest that deep-learning-based approaches are very suitable for time-series forecasting [5,17–21]. In [22], the authors suggested that, in several

application domains, deep learning algorithms dominate machine learning approaches and have a superior ability to learn the non-linear structure between the input and output.

Long Short-Term Memory (LSTM) is one of the deep-learning models, specifically designed to handle sequential data. GRU is a more recent addition to deep-learning-based sequence models, and it simplifies the architecture of LSTM to an extent. In recent papers [5,8,9,23–25], the authors used traditional unidirectional LSTM and GRU to forecast solar irradiation, and they demonstrated that both LSTM and GRU dominate standard statistical techniques as well as state-of-the-art machine-learning models. For time-series data, LSTM is most popularly used for intra-hour (short-term) solar forecasting [26]. Subsequently, in some of the recent research papers [27,28], the authors demonstrated that, by replacing traditional LSTM with bidirectional LSTM, they produced better forecasting performance for the energy domain.

A bidirectional deep-learning model is a combination of two sequential layers: one layer is trained with the preceding values, $(t - k)$ to $(t - 1)$ th term of the sequence, referred to as the past context to predict the t th term. This is typically called the forward layer. The other layer uses $(t + 1)$ to $(t + k)$ th term of the sequence referred as the future context to predict the t th term. This is known as the backward layer. The above process is described by taking the window size of k . Both these contexts are available when we are predicting missing values or finding representation of words by context. Understandably for a typical forecasting task, the future context is unavailable. In some of the recent works [27,28], the authors used the same past context as both the past and future context and reported better results.

In this paper, we propose a feature representation, called a bidirectional feature, that augments the past context and uses a simple technique to construct the future context. This representation is further discussed in subsequent sections.

The main contributions of this paper are summarized as follows:

- Bidirectional GRU is applied for the first time to solar energy forecasting, and it is shown to be better performing than other common sequence models, such as unidirectional LSTM, Bidirectional-LSTM (BLSTM), and Unidirectional GRU.
- A new feature representation with a bidirectional nature is proposed, which further augments the performance of BGRU. The model shows improved performance compared to two state-of-the-art models.
- The performance of the model is validated on real-life data from six solar stations from three climatic zones and in two seasons in India.

The rest of the paper is organized as follows. In Section 2, the recent forecasting models for renewable energy are outlined. In Section 3, the detailed architecture of BLSTM and BGRU with the proposed bidirectional feature representation are elaborated. In Section 4, the materials and methods employed in setting up the empirical study is discussed. In Section 5, the results of the prediction models are presented along with critical analysis and discussion. Finally, in Section 6, our concluding remarks are presented.

2. Forecasting Models for Renewable Energy

In this section, some of the recent research efforts in the domain of renewable energy forecasting are presented. They are classified broadly in the following two categories, namely (a) machine-learning-based models and (b) deep-learning-based models.

2.1. Machine Learning Based Models

In [10], the authors reported that, compared to SVR, decision tree regressor and k-nearest neighbors (kNN), Multilayer Perceptron (MLP) performed the best to forecast solar irradiation for one-hour-ahead prediction. In [29], the authors proposed a unique re-sampling technique to design a uni-variate solar PV power forecasting model using machine learning algorithms for a forecasting horizon of length of five minutes to three hours.

Out of Multi-Layer Perceptron (MLP), Support Vector Regression (SVR), Random Forest (RF), and Multiple Linear Regression (MLR), RF was reported to have the best

accuracy compared to the others. In [30], the authors predicted three components of hourly solar irradiation, namely global horizontal, beam normal, and diffuse horizontal for a horizon of 6 h. The authors used three different models, namely smart persistence, ANN and RF. They observed that RF showed the best performance for all three components of solar irradiation.

Some of the approaches can be classified as hybrid using both ANN and ML. As example, In [31] the authors used a radial basis neural network for solar power prediction in the Netherlands, where the parameters were tuned using swarm optimization. In paper [32], the authors advocated the use of an Ensemble model for both wind and solar energy prediction as they can reduce variance of the base learners. It can be observed that most of the research has been conducted for short-term solar forecasting [10,29–31]. Currently, RF has been popularly used in certain studies [29,30,32] for short-term solar forecasting.

2.2. Deep-Learning-Based Models

In [5], the authors used LSTM to implement a solar irradiation prediction model. They reported that LSTM outperformed other models, such as ANN, support vector machines (SVM), and autoregressive moving average (ARMA). In [9], the authors proposed a unidirectional LSTM-based day-ahead hourly solar forecasting model to forecast irradiation along with other meteorological features. In [8], an hourly univariate photovoltaic power forecasting model was proposed based on LSTM-RNN.

In [33], the authors reported a day-ahead hybrid PV power forecasting model, where the authors used LSTM and Convolutional neural network (CNN) to build the model. In [34], the authors observed that the LSTM model does not need pre-processing and works best with stateful models. In deep-learning-based approaches, there are also approaches based on CNN, where the authors attempted to focus on the global information unlike the sequence models [35].

It can be observed that, currently, unidirectional LSTM has proved its excellence over many state-of-the-art machine learning as well as statistical models in the context of short-term solar forecasting [5,9,33]. In paper [27], the authors developed a hourly PV power forecasting model. In this context, the authors studied the effectiveness of different time series prediction models divided into two classes statistical (ARMA, ARIMA, SARIMA) and Neural Network (NN)-based models (LSTM, Bidirectional-LSTM).

It has been shown that NN-based models, specifically Bidirectional-LSTM outperformed others. They concluded that, for one-hour-ahead prediction, NN-based models performed considerably well. In [28], the authors proposed a solar power prediction model based on different variants of LSTM and as well as different Bidirectional LSTM variants with two choices of architectures single layer and double layer. The authors claimed that bidirectional LSTM of a single layer outperformed others and also they validated their proposed model against four different seasons.

In Table 1, we summarize the above-discussed research works in terms of several attributes, such as the type of the research data, proposed technique, length of the forecasting window, the country, correctness, and advantages or disadvantages. Recent papers adopted Bidirectional LSTM and reported better results. Nevertheless, we observed that Bidirectional GRU were not applied, and the same set of observations were used for both forward and backward context.

Table 1. Studies based on currently implemented models on solar irradiation prediction.

Citation	Data	Model Name	Forecasting Window	Country	Correctness	A/D
[8]	Time Series	LSTM (Unidirectional)	Hourly	Egypt	The claimed forecasting error is 82.15, and 136.87 in terms of RMSE	Performed better compared to MLR, BRT, and NN
[27]	Time Series	Bi-LSTM (Bidirectional)	Hourly	China	Bi-LSTM produced correlation coefficient of 98%, and RMSE of 0.791	Same past context used for both the forward and the backward mode
[28]	Time Series	PSO-LSTM (Bidirectional)	Multiple days	China	PSO-LSTM achieved the lowest MAE, and RMSE as 8.14, and 19.41	Same past context used for both the forward and the backward mode
[36]	Time Series	CNN-LSTM (Unidirectional)	1-Day, 1-Week, 2-Week and 1-Month	Australia	It achieved lower MAPE < 11%, and RRMSE < 15% compared to benchmark models	This study is limited to one solar station
[33]	Time Series	LSTM-CNN (Unidirectional)	Multiple days	China	LSTM-CNN achieved the best MAE, RMSE, and MAPE as 0.221, 0.621, and 0.042	This study is limited to one solar station
[6]	Time Series	MLP (Unidirectional)	Monthly	UAE	MLP has shown the best MBE, and RMSE as 0.0003, and 0.179	The model is validated for three solar stations
[30]	Time Series	RF	1 h to 6 h	France	RF achieved the lowest forecasting error as 19.65% to 27.78% in terms of RMSE	This study is restricted to one solar station
[10]	Aerosol Optical Depth (AOD) and the Angstrom Exponent data	MLP (Unidirectional)	1 h	Saudi Arabia	MLP achieved lower RMSE under 4% and forecast skill of over 42%	The study is restricted to one solar site
[29]	Time Series	RF	5 min to 3 h	Australia	RF achieved the lowest overall MAE, and MRE as 110.46, and 10.5%	The proposed model is univariate, and restricted to one solar site

3. Detailed Working of the Deep-Learning-Based Models

This section has two subsections. In the first part, a description of the sequence models is provided. In the second part, the different feature representation options for the sequence models are elaborated. In Section 3.2, the proposed augmented bidirectional feature representation is explained, which is one of the main contributions of the paper.

3.1. Sequential Deep Learning Models

Sequence or ordered data is present in various application domains namely, climate, finance, medical diagnosis, astronomy, bio-informatics, etc. Traditional machine-learning models consider all features to be independently and identically distributed (iid), which do not apply here. Recurrent Neural Networks (RNN) are deep-learning models designed to handle sequence data. In RNN, the next step output is dependent on the current input as well as the previous step output.

This was much better than sequence data but was prone to a problem known as the Vanishing Gradient problem. As a result of which, RNN was incapable of remembering long-term dependency. LSTM [37], is a special type of RNN, where the hidden memory state update is customized through some unique gates. Hence, LSTM shows a better understanding in the case of long-range dependencies and is effective with the vanishing [38] and exploding gradient problem [39].

GRU [40] is mostly similar to LSTM and was proposed by [41]. However, it has some specific advantages over LSTM. GRU is less complicated than LSTM, and it uses fewer parameters compared to LSTM. Hence, it is faster than LSTM. LSTM uses both input and forget gate while GRU performs both these operations using the reset gate. The function

of the reset gate is to decide how to combine the current input with the old memory. The update gate decides what amount of previous memory needs to maintain. The equations that govern the working of GRU are given as follows:

$$\text{Reset gate}(r_t) = \sigma(W_r h_{t-1} + U_r X_t) \quad (1)$$

$$\text{Update gate}(z_t) = \sigma(W_z h_{t-1} + U_z X_t) \quad (2)$$

$$\text{Cell state}(c_t) = \tanh(W_c (r_t \circ h_{t-1}) + U_c X_t) \quad (3)$$

$$\text{New state}(h_t) = (z_t \circ c_t) + ((1 - z_t) \circ h_{t-1}) \quad (4)$$

Here, X_t is the input vector at timestamp t , and σ denotes the sigmoid activation function. h_t and h_{t-1} are the hidden state vectors. W_r , W_z and W_c are the parameter matrices related to hidden state vector h_{t-1} for reset gate, update gate, and the current cell state. U_r , U_z , and U_c are the parameter matrices related to the input vector X_t for reset gate, update gate and the current cell state. \circ implies element-wise matrix multiplication, and the new state h_t is the final output vector.

- The GRU model, rather than any sequential model, is trained by selecting a continuous portion or window from the input data. Instead of taking all such windows for training, it is often broken into batches.
- If the batches are considered dependent on each other, then it is called a stateful model.
- Typically, when dealing with the sequence data, the hidden layer nodes are any sequential cells. In Figure 1, a simple schematic diagram of a deep neural network is shown, whereas a basic building block in the hidden layers, the GRU cells are used. The inputs and the outputs are denoted as $[I_1, I_2, I_3, \dots, I_n]$, and $[O_1, \dots, O_m]$ respectively.
- As with traditional neural networks, gradient-descent and back-propagation are used to learn the parameters of the network. Some of the state-of-the-art optimizers are ADAM, RMSProp, Stochastic Gradient Descent [42–44], etc.

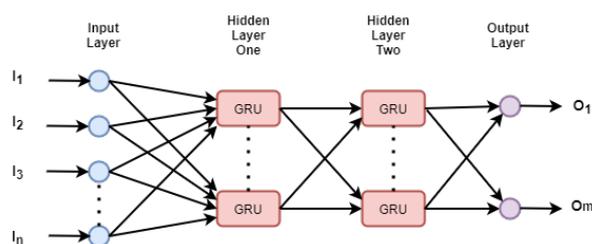


Figure 1. A neural network based on GRU cells.

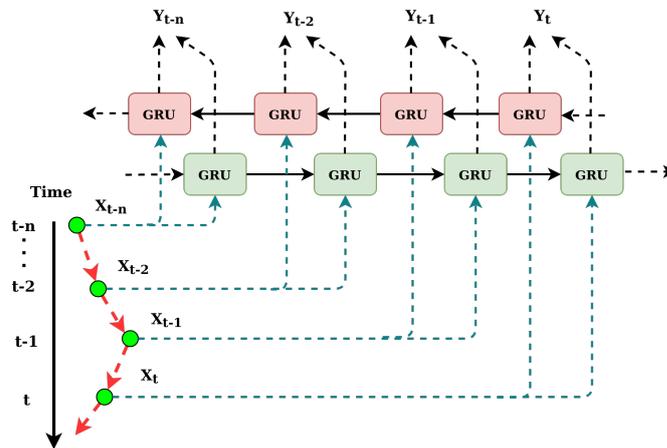
In the sequence, a few preceding values of the sequence were used to predict the current value. For tasks, such as missing data prediction, suitable text representation, speech recognition, etc., instead of only the preceding sequence, the succeeding sequence data is available. BLSTM [45,46] and BGRU are specifically designed for the same. In Figure 2a, the detailed generic architecture of BGRU is shown where the length of the forward and backward context is identical. In Figure 2b, the length of the forward and backward context is different.

The functionality of BGRU is as follows:

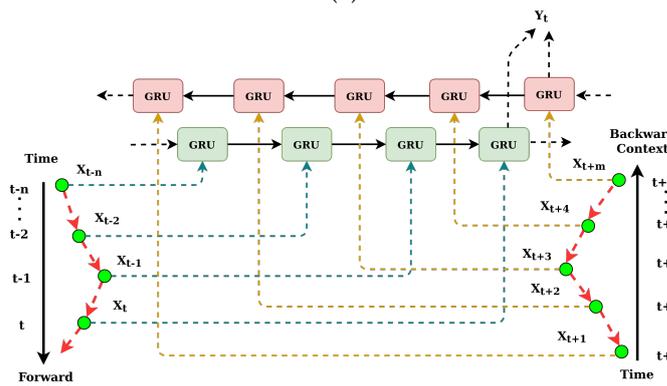
- BGRU is a fusion of two independent unidirectional GRU layers when one layer maintains the forward hidden states whereas the other maintains the backward hidden states. In the forward pass, BGRU processes inputs sequence as $\dots, X_{t-3}, X_{t-2}, X_{t-1}, X_t$ for the time steps $\dots, t-3, t-2, t-1, t$, and, in the backward pass,

BGRU processes the input sequence for the time steps $t, t - 1, t - 2, t - 3, \dots$ in the reverse direction.

- After both forward and backward passes were completed, the hidden states are concatenated to form a final single set of hidden states.
- Then, the final hidden states go through a densely connected layer to produce the output sequence as $\dots, y_{t-3}, y_{t-2}, y_{t-1}, y_t$.



(a)



(b)

Figure 2. A network of bidirectional GRU (a) contexts with similar lengths and (b) contexts with different length.

3.2. Feature Preparation for Sequential Models

In this section, four different types of feature representations are explained. The models trained with these different feature representations are named ULSTM (Unidirectional LSTM + Unidirectional Feature), M-ULSTM (Unidirectional LSTM + Augmented Feature), BLSTM, BD-BLSTM (Bidirectional LSTM + Proposed Bidirectional Feature), and BD-BGRU (Bidirectional GRU + Proposed Bidirectional Feature). In Table 2, the models are compared based on several attributes, such as type of the model, feature type, and the input feature representation, and also we explain the generic architecture of each model in the following:

1. **ULSTM:** In Figure 3a, the high-level block diagram of ULSTM is presented. In this case, to predict solar irradiation for the time step t as X_t , usually, the past context of the same day is used. For example, to predict X_t , the input sequence is defined as $[X_{t-m}, \dots, X_{t-2}, X_{t-1}]$. The traditional sliding window approach was used to represent the whole feature set. The length of the window is denoted as m .

2. **M-ULSTM:** As with the previous feature representation, this also uses past values, and hence is a unidirectional model. However, the past context is augmented here, with the previous values, corresponding to the same time. In Figure 3b, the high-level block diagram of M-ULSTM is presented. For example, for predicting solar irradiation for the time step t as X_t for Day_0 we not only used the same day past sequence denoted as $Day_0[X_{t-m}, \dots, X_{t-2}, X_{t-1}]$ but also values from previous day denoted as $Day_{-1}[X_{t-m}, \dots, X_{t-2}, X_{t-1}]$.
3. **BLSTM:** This is the traditional variant of bidirectional LSTM, where the same context is used to train the model both from forward and backward directions. Figure 3 b depicts the block diagram of BLSTM. In this architecture, the same augmented features were used in both the left and right contexts.
4. **BD-BLSTM and BD-BGRU:** In the case of time-series prediction problems, such as text data, traditional bidirectional deep-learning models use the same past sequence for both the forward and backward context. In this paper, a simple technique is proposed, first to augment the past context and next to construct future context from previous day. The model used here is more generalized allowing the past and future context to be of unequal length. In Figure 3d, the block diagram of this proposed bidirectional feature set with BLSTM and BGRU is presented. The right or backward context is collected from the previous day denoted as $Day_{-1}[X_{t+1}, \dots, X_{t+m-2}, X_{t+m-1}]$.

Table 2. Comparison of models with different input feature representations.

Models	Input Sequence	Remarks
ULSTM	$Day_0[X_{t-m}, \dots, X_{t-2}, X_{t-1}]$	Same day m input time steps
M-ULSTM	Augmented Feature $\{Day_{-1}[X_{t-m}, \dots, X_{t-2}, X_{t-1}]$ $Day_0[X_{t-m}, \dots, X_{t-2}, X_{t-1}]\}$	Same day m input time steps are augmented with previous day m time steps
BLSTM	$L_{context} =$ Augmented Feature $\{Day_{-1}[X_{t-m}, \dots, X_{t-2}, X_{t-1}]$ $Day_0[X_{t-m}, \dots, X_{t-2}, X_{t-1}]\}$ $R_{context} =$ Augmented Feature $\{Day_{-1}[X_{t-m}, \dots, X_{t-2}, X_{t-1}]$ $Day_0[X_{t-m}, \dots, X_{t-2}, X_{t-1}]\}$	Same day m input time steps along with previous day m time steps is used as both forward and backward context
BD-BLSTM/ BD-BGRU	$L_{context}$ Augmented Feature $\{Day_{-1}[X_{t-m}, \dots, X_{t-2}, X_{t-1}]$ $Day_0[X_{t-m}, \dots, X_{t-2}, X_{t-1}]\}$ $R_{context}$ $Day_{-1}[X_{t+1}, \dots, X_{t+m-2}, X_{t+m-1}]$	Same day m input time steps along with previous day m time steps is used as forward context and augmented with previous day m future time steps as backward context

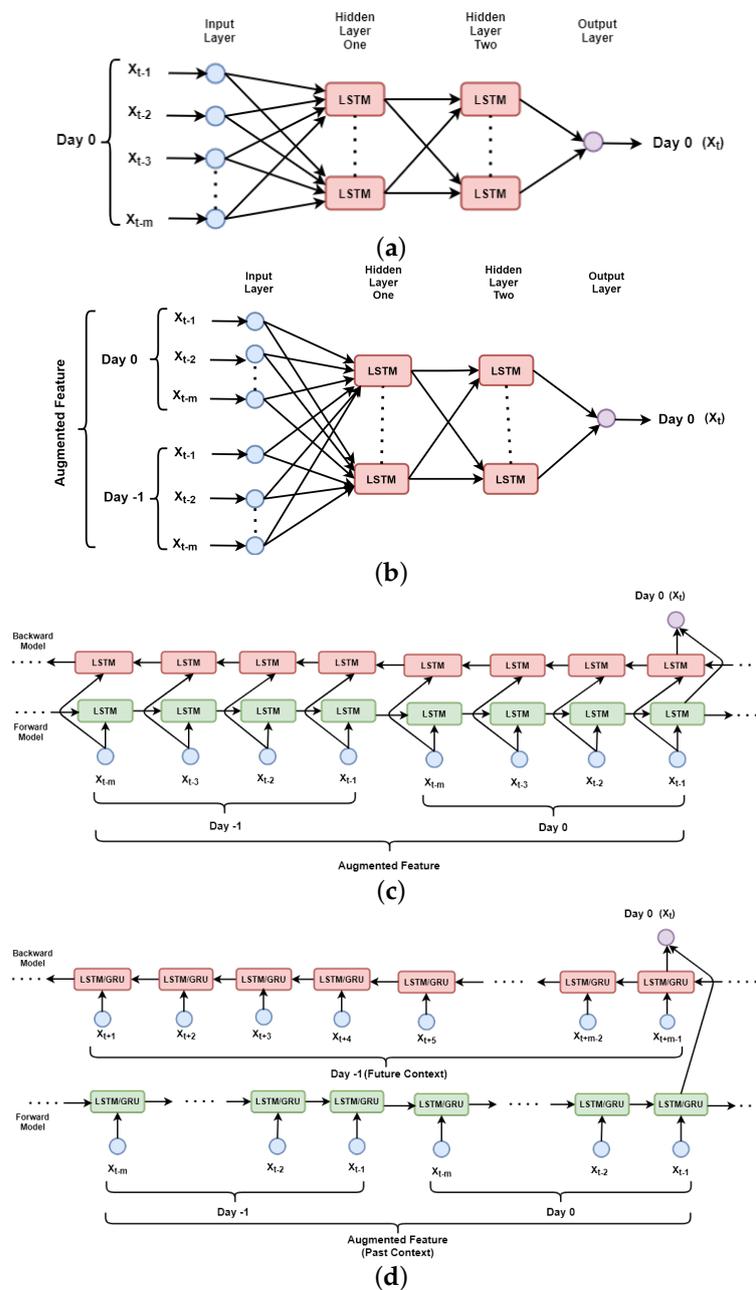


Figure 3. Schematic diagram of (a) ULSTM (b) M-ULSTM (c) BLSTM (d) BD-BLSTM/BD-BGRU.

4. Materials and Methods

This section has five subsections. In the first subsection, how the data is collected and its different characteristics are described. In the second subsection, the pre-processing steps are discussed that are applied to the data. In the third section, the experimental setup of the models is discussed. In the fourth section, the reference models are briefly discussed, and also they been used for comparison, and finally, in the fifth subsection, various performance metrics are enclosed that are employed for evaluation and comparison of the models.

4.1. Data-Set Description

We used the application programming interface (API) provided by the Center for Wind Energy Technology (C-WET) to crawl solar irradiation data for Solar Radiation Resource Assessment (SRRA) stations across India. In this paper, data from 2016 is used for three climatic zones (Hot and Dry, Warm and Humid, and Cold and Cloudy), and

six stations located at Chennai (Tamil Nadu), Howrah (West Bengal), Guntur (Andhra Pradesh), Kotada Pitha (Gujrat), and Ajmer (Rajasthan).

Figure 4, depicts solar stations in different climatic zones. Table 3 describes the details of the solar stations, location, etc. For each of the stations, we chose a month each from the rainy and winter season because the rainy season is known for its high variability due to cloud and rain, on the other hand, in winter comparatively, the uncertainty is less. C-WET provides data with a one-minute resolution. In our work, this was aggregated into a five minute resolution.

Table 3. Description of the data.

City with Month	Longitude	Latitude	Standard Deviation	Climatic Zone
Ajmer (January)	74.66° E	26.40° N	47.64	Hot and dry
Chennai (December)	80.22° E	12.96° N	48.26	Warm and humid
Dehradun (January)	77.97° E	30.42° N	46.94	Cold and cloudy
Guntur (January)	80.53° E	16.37° N	49.46	Warm and humid
Howrah (December)	88.31° E	22.55° N	45.16	Warm and humid
Kotada Pitha (January)	71.21° E	21.95° N	48.55	Hot and dry
Ajmer (August)	74.66° E	26.40° N	78.33	Hot and dry
Chennai (November)	80.22° E	12.96° N	51.88	Warm and humid
Dehradun (August)	77.97° E	30.42° N	74.78	Cold and cloudy
Guntur (August)	80.53° E	16.37° N	90.36	Warm and humid
Howrah (August)	88.31° E	22.55° N	80.22	Warm and humid
Kotada Pitha (August)	80.53° E	16.37° N	82.91	Hot and dry

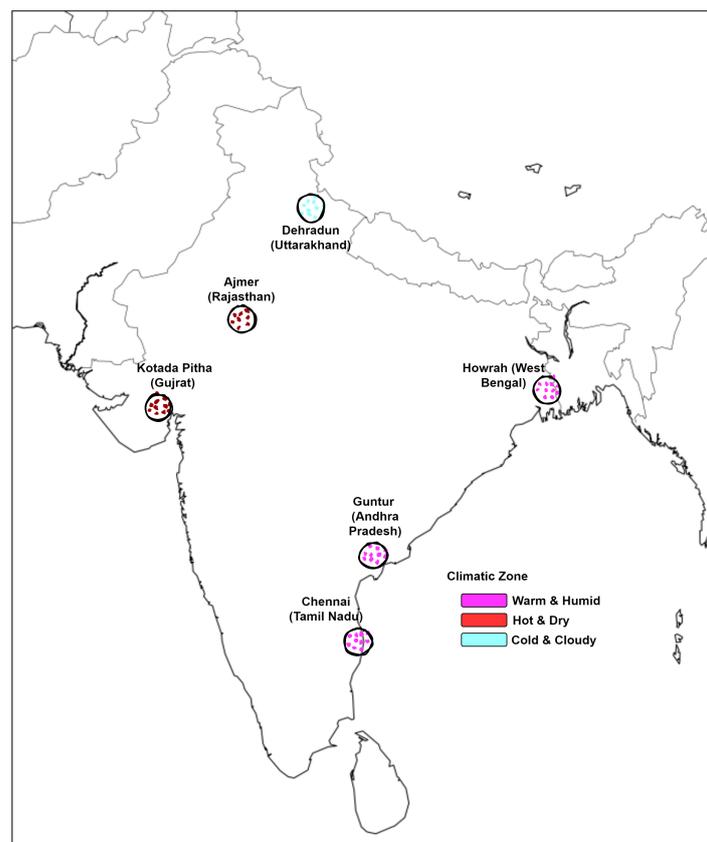


Figure 4. Solar stations based on climatic zones.

4.2. Data Pre-Processing

Following pre-processing on the raw data [47]:

1. For each station–month combination, to remove night hours, only the measurements of GHI between 7 a.m. to 7 p.m. were used.
2. After that, the GHI values for each day were aggregated into five minutes, and then merged for all the days in a single time-series. The time-series should be formatted as a three-dimensional array, where the three dimensions are the size of the batch, number of time-steps (Window Size), and number of input features. In a single window, 20 time-steps of GHI were used. Batch size refers to the number of training samples used at the time of the training phase for one iteration. We used 100 training samples in a batch. During the learning process, successive batches are used to train the network.
3. Finally, GHI values were normalized in between $[-1,1]$ using the following transformation (5).

$$\widehat{X}_t = \frac{X_t - X_{min}}{X_{max} - X_{min}} \quad (5)$$

4.3. Experimental Setup

To predict GHI 20 steps (1 h 40 min) ahead, historical solar irradiance data was used. The whole time-series data is decomposed into train and test set with the proportion of 80% and 20%. A subset of training data is used as a validation set (20% of the training set) to validate the model. In this work, different variants of LSTM and GRU architectures were implemented, and the prediction performance is compared against two recent benchmark models. In the following section, the technical details of all implemented models are described briefly.

- Different architectures of LSTM and GRU were developed using the Keras [48] API in python. For bidirectional models, three hidden layers, and for the traditional LSTM, two hidden layers were used.
- In the input layer of both LSTM and GRU, different choices of input size, i.e., sequential length were varied from 20 to 60 steps.
- In the output layer, the 20-time steps were predicted, which is 100 min in this case.
- For the bidirectional models, two sequential models were applied separately. Prediction were made from the forward direction by one model and from the backward direction by the other model. Finally, by combining both predictions, the actual decision is made.
- In the output layer, 20 neurons with linear activation were used. For traditional LSTM, the same non-linear activation tanh [49] was used, and in the output layer, linear activation was used.
- All models are trained on Adaptive Moment Estimation (Adam) [50] optimizer. For all the implemented models, different hyper-parameters [51], such as learning rate, the number of nodes in different hidden layers, batch size, and the number of epochs were optimized using Bayesian Optimization [52] approach. In this context, the Tree-structured Parzen Estimator (TPE) [53] algorithm of Hyperopt [54] package in python is used. In Table 4, the details of all the hyper-parameters were enlisted.

Table 4. Hyper-parameters to optimize.

Models	Hyper-Parameters	Values
BLSTM/BD-BLSTM/BD-BGRU/ULSTM/M-ULSTM	Number of hidden layers	1, 2, 3
	Nodes in hidden layer	25, 50, 100
	Learning rate	0.1, 0.01, 0.001
	Batch size	1, 10, 20, 50, 100
	Epoch	20, 40, 60, 60, 80, 100

4.4. Comparison with Other Models

The prediction performance of the proposed methods BD-BLSTM and BD-BGRU are compared with two recently published works on solar power forecasting. In the first reference model [29], the authors used RF from the SciKit-learn package [55].

They reported a multi-step ahead prediction model, where RF was applied for each output step individually. It is to be noted, for each RF model, separately, multiple hyper-parameters, such as `n_estimators` (number of trees in the forest), `min_samples_split` (minimum number of samples required to split an internal node), and `min_samples_leaf` (minimum number of samples required to be at a leaf node) were tuned using the Grid Search [56] approach with 10-fold cross-validation. To represent input features for each output step separately, they proposed a unique resampling technique for each RF model. In [8], the authors used the traditional LSTM with two sequential layers to design the forecasting model.

In Table 5, the architectural details of all the reference models and the proposed model are presented. From the table, some notable findings can be made, which are listed as follows:

- The reference models [8,29], used univariate time series data, such as the proposed model.
- In [8], the authors used raw time series (Non-stationary) to design the forecasting model, similar to this work.

Table 5. Architectural similarity and dissimilarity of the proposed model compared to the reference models.

Model Attributes	[29]	[8]	BD-BLSTM/BD-BGRU *
Similarities			
Domain:	Solar prediction	Solar prediction	Solar prediction
Input data:	Time series	Time series	Time series
Prediction type:	Uni-variate	Uni-variate	Uni-variate
Series type:	Non-stationary	Non-stationary	Non-stationary
Dissimilarities			
Model type:	Random Forest	Neural Network	Neural Network
Sequential model:	×	✓	✓
Memory:	×	✓	✓
Stateful:	×	×	✓
Activation:	×	Default	tanh
Bidirectional:	×	×	✓
Bidirectional feature:	×	×	✓
Stateful:	×	×	✓

4.5. Performance Metrics

In Equation (6), the computation of the Normalized Root Mean Square Error (nRMSE) is shown. nRMSE is known for its scale independence property. GHI_t is the t th actual value, and the corresponding predicted value is represented as \widehat{GHI}_t . This always returns a non-negative value. The closer the value to zero, the better is the fit of the model with the data.

$$nRMSE = \frac{\sqrt{\frac{\sum_{i=1}^n (GHI_t - \widehat{GHI}_t)^2}{n}}}{\sum_{i=1}^n GHI_t} \quad (6)$$

5. Results and Discussions

This section has five subsections. In the first subsection, the prediction performance of M-ULSTM is analyzed compared to ULSTM. After that, in the second subsection, the prediction performance of BLSTM is compared to M-ULSTM. In the third subsection, the

forecasting accuracy of BD-BLSTM and BD-BGRU is compared against BLSTM. Then, in the fourth subsection, the overall forecasting performance of BD-BGRU is presented. Finally, in the fifth subsection, the forecasting performance of the proposed model, namely BD-BGRU is compared with some recent works.

5.1. Forecasting Performance of M-ULSTM over ULSTM

In this section, the forecasting performance of M-ULSTM is compared with ULSTM based on nRMSE. In Table 6, the season-specific nRMSE scores for all the stations are presented. The model-specific standard deviation is included in the table to analyze the prediction variability.

Winter: For all the stations, on average M-ULSTM outperformed ULSTM by 83.28%.

Rainy: ULSTM outperformed by M-ULSTM by 38.87%.

Variability: M-ULSTM achieved a much lower standard deviation than ULSTM.

Table 6. The forecasting performance of M-ULSTM and ULSTM measured on nRMSE.

Models	Ajmer		Chennai		Dehradun		Guntur		Howrah		Kotada Pitha		σ
	Winter	Rainy											
ULSTM	0.042	0.103	0.035	0.064	0.042	0.056	0.059	0.212	0.023	0.158	0.031	0.131	0.0581
M-ULSTM *	0.010	0.064	0.001	0.017	0.007	0.039	0.012	0.185	0.001	0.083	0.010	0.121	0.038

5.2. Forecasting Performance of BLSTM over M-ULSTM

In this section, the overall season specific forecasting performance of BLSTM is compared against M-ULSTM. In Table 7, for all station–season combinations, the forecasting performance is calculated based on nRMSE. The model-specific standard deviation is computed.

Winter: We observed that BLSTM outperformed M-ULSTM by 11.64%.

Rainy: In the rainy season, BLSTM dominated M-ULSTM by 59.53%.

Variability: BLSTM achieved a lower standard deviation in its prediction compared to M-ULSTM.

Table 7. The forecasting performance of BLSTM and M-ULSTM measured on nRMSE.

Models	Ajmer		Chennai		Dehradun		Guntur		Howrah		Kotada Pitha		σ
	Winter	Rainy											
M-ULSTM	0.010	0.064	0.001	0.017	0.007	0.039	0.012	0.018	0.001	0.083	0.010	0.121	0.038
BLSTM *	0.009	0.057	0.002	0.015	0.007	0.039	0.014	0.147	0.001	0.056	0.012	0.078	0.025

5.3. Forecasting Performance of BD-BLSTM and BD-BGRU over BLSTM

In Table 8, the forecasting performance of BD-BLSTM and BD-BGRU is compared against BLSTM, where the forecasting accuracy is measured on nRMSE.

- **BD-BLSTM:**

Winter: In the winter season, BD-BLSTM dominated BLSTM by 4.04%.

Rainy: In the rainy season, BD-BLSTM outperformed BLSTM by 31.81%.

Variability: BD-BLSTM achieved lower standard deviation compared to BLSTM.

- **BD-BGRU:**

Winter: In the winter season, BD-BGRU outperformed BLSTM by 21.49%.

Rainy: In the rainy season, BD-BGRU outperformed BLSTM by 53.45%.

Variability: BD-BGRU achieved a lower standard deviation compared to BLSTM.

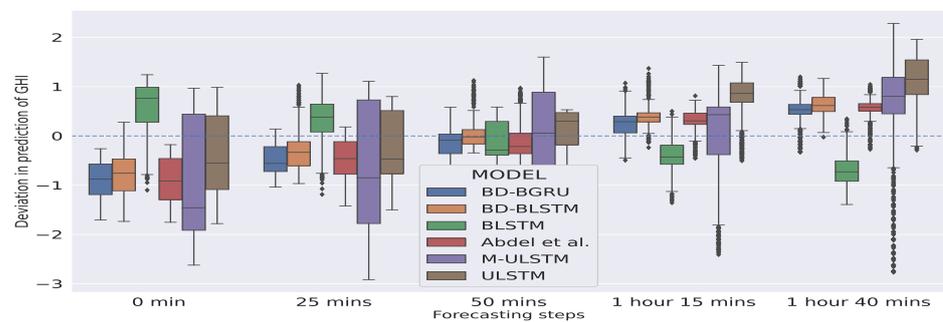
Hence, the above experimental result suggested that, when the proposed bidirectional feature representation was used, BD-BLSTM and BD-BGRU can forecast GHI with higher accuracy compared to BLSTM.

Table 8. The forecasting performance of BLSTM, BD-BLSTM, and BD-BGRU measured on nRMSE.

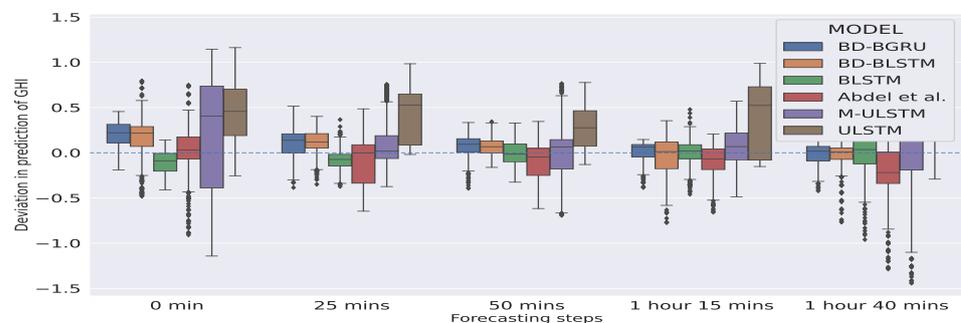
Models	Ajmer		Chennai		Dehradun		Guntur		Howrah		Kotada Pitha		σ
	Winter	Rainy											
BLSTM	0.009	0.057	0.002	0.015	0.007	0.039	0.014	0.147	0.001	0.056	0.012	0.078	0.025
BD-BLSTM	0.009	0.0029	0.003	0.0017	0.007	0.023	0.014	0.068	0.001	0.038	0.011	0.056	0.021
BD-BGRU *	0.008	0.024	0.001	0.012	0.006	0.014	0.010	0.061	0.001	0.026	0.009	0.026	0.016

5.4. Overall Forecasting Performance of BD-BGRU

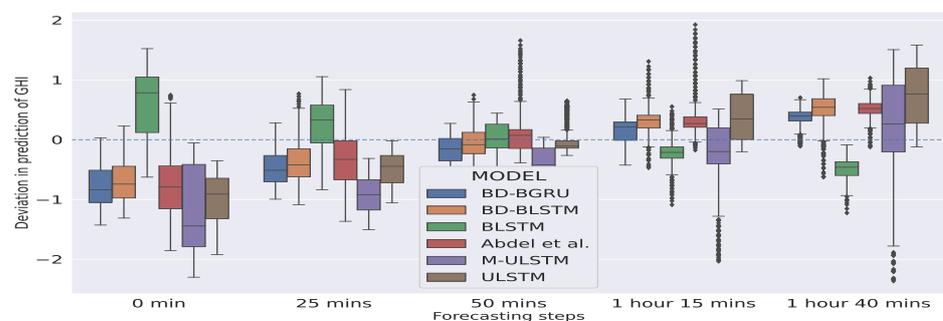
In Figures 5 and 6, the season-specific variability of the deviation of GHI is presented for each station-month combination. The variability was calculated using box-plot and is plotted over different choices of forecasting horizons. The deviation of GHI was calculated by subtracting the predictions from the actual values. We observed that, in winter, for each forecasting horizon, BD-BGRU achieved the lowest variability in the deviation of GHI.



(a) Ajmer-January

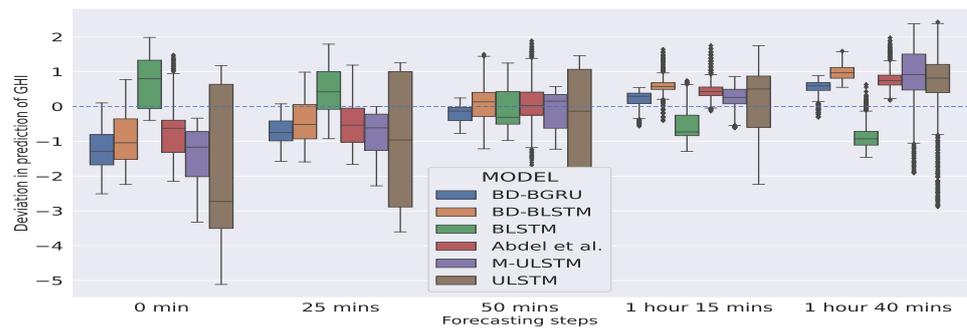


(b) Chennai-December

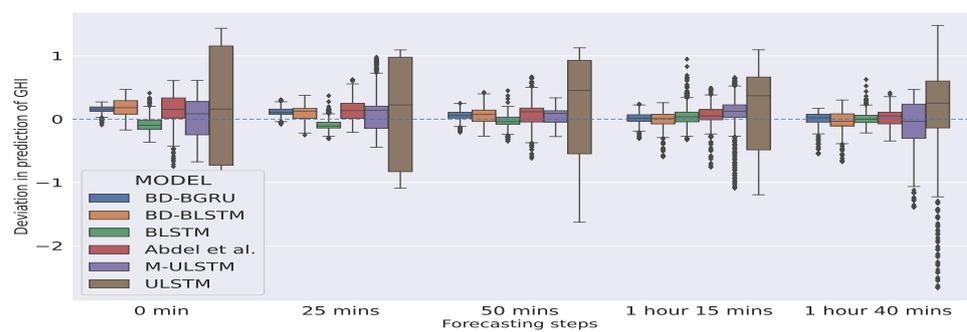


(c) Dehradun-January

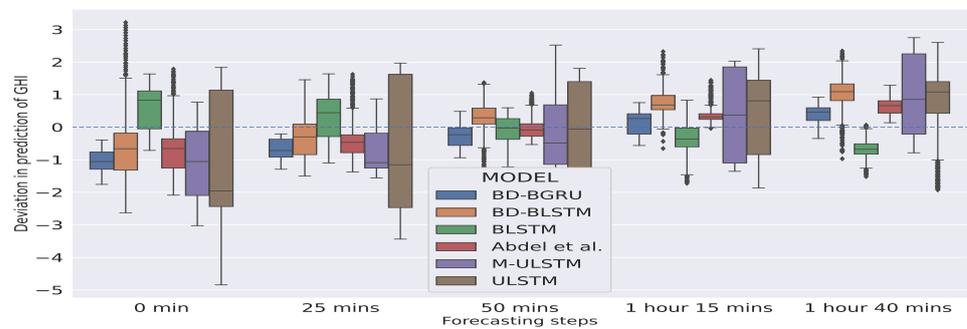
Figure 5. Cont.



(d) Guntur-January



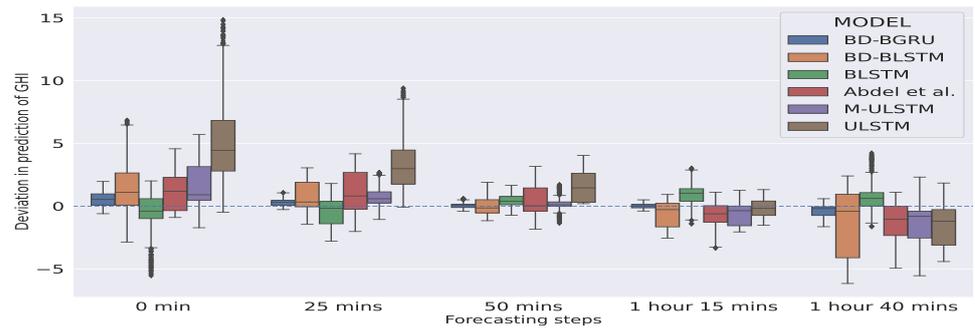
(e) Howrah-December



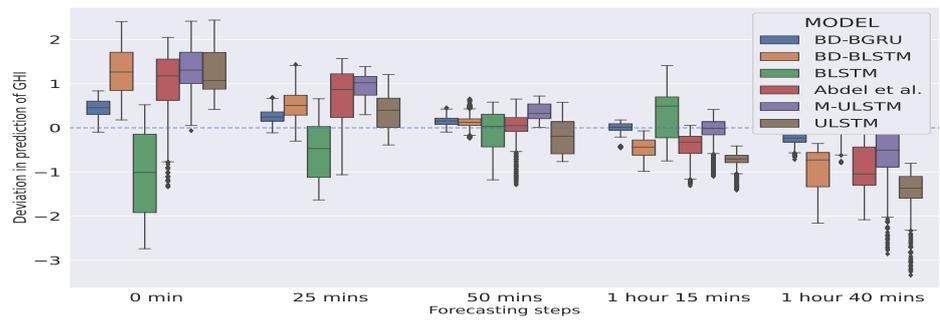
(f) Kotada Pitha-January

Figure 5. Variability in deviation of GHI (W/m^2) in the winter season.

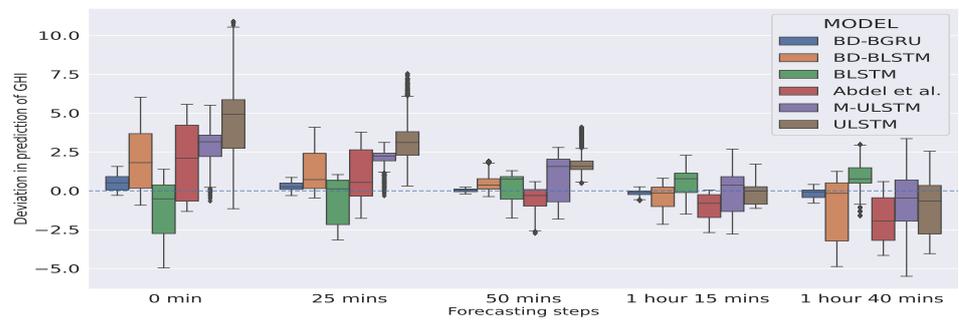
Similarly, in the rainy season, BD-BGRU achieved the lowest variability in its deviation of GHI compared to other models. ULSTM has performed the worst for both winter and rainy seasons. Furthermore, compared to other models for all forecasting horizons, BD-BGRU achieved much lower divergence in its predictions compared to the actual results.



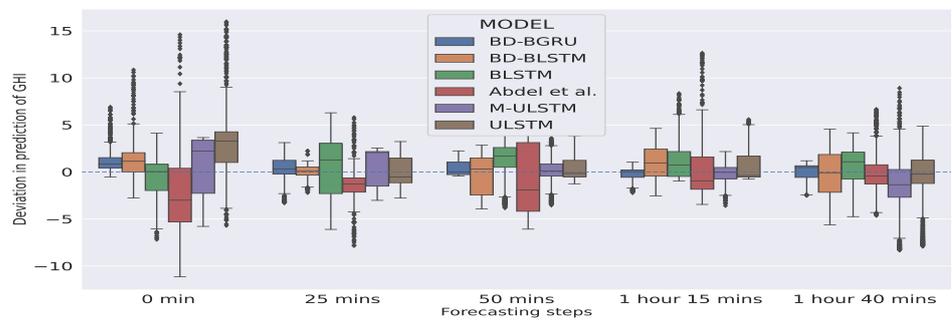
(a) Ajmer-August



(b) Chennai-November

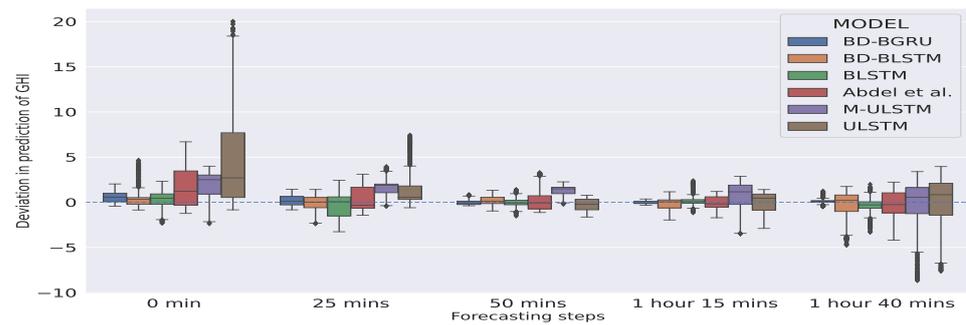


(c) Dehradun-August

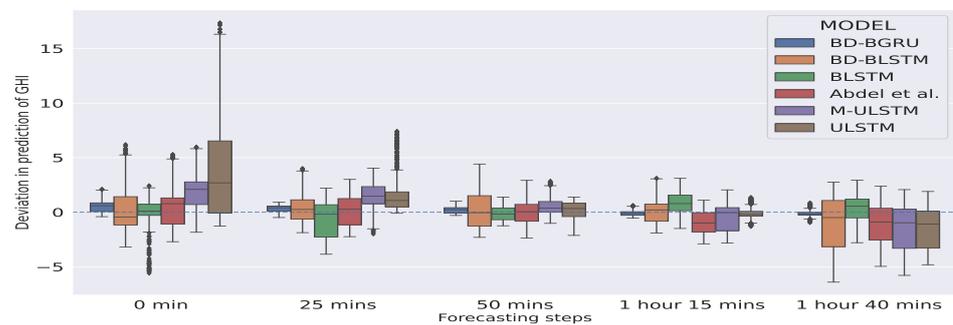


(d) Guntur-August

Figure 6. Cont.



(e) Howrah-August



(f) Kotada Pitha-August

Figure 6. Variability in deviation of GHI (W/m^2) in the rainy season.

5.5. Comparison with Other Models

In this section, the overall forecasting performance of BD-BGRU is compared against the benchmark models. Table 9 depicts the forecasting performance of BD-BGRU compared to the benchmarks in terms of nRMSE. The summary of this paper is presented in Table 10.

Winter: In winter season, BD-BGRU outperformed [8,29] by 91.17% and 74.93%.

Rainy: In rainy season, [8,29] were outperformed by BD-BGRU by 82.48% and 58.61%. Compared to the benchmark models, BD-BGRU achieved the lowest standard deviation of 0.016.

Table 9. The forecasting performance of BD-BGRU compared to the benchmark models.

Models	Ajmer		Chennai		Dehradun		Guntur		Howrah		Kotada Pitha		σ
	Winter	Rainy											
[29]	0.057	0.131	0.130	0.128	0.057	0.134	0.075	0.180	0.157	0.162	0.066	0.153	0.043
[8]	0.026	0.044	0.166	0.015	0.025	0.166	0.025	0.195	0.004	0.062	0.030	0.081	0.067
BD-BGRU *	0.008	0.024	0.001	0.012	0.006	0.014	0.010	0.061	0.001	0.026	0.009	0.026	0.016

Figures 7 and 8 depict the season and climatic-zone specific variability in predictions of BD-BGRU and compared with other forecasting models. Figure 7 shows the season-specific box-plot. We observed that, in winter, the variability in predictions of BD-BGRU was the lowest. Moreover, compared to BD-BGRU, BD-BLSTM, BLSTM, and M-ULSTM have similar forecasting performances in winter. However, in the rainy season, the variability in predictions of BD-BGRU is much lower than the other models. Hence, the above discussion suggested that, in winter, BD-BGRU, BD-BLSTM, BLSTM, or M-ULSTM can be alternatively used. However, in the rainy season, BD-BLSTM demonstrated superior forecasting performance.

Figure 8 shows the climatic-zone specific box-plots of the nRMSE scores of all the models. We observed that, in all climatic zones, BD-BGRU achieved the lowest variability

in prediction compared to the benchmarks as well as other forecasting models. Moreover, in hot and dry and warm and humid climatic zones, the prediction variability of BD-BGRU is more stable. In all climatic zones, BD-BLSTM is the second best performing forecasting model.

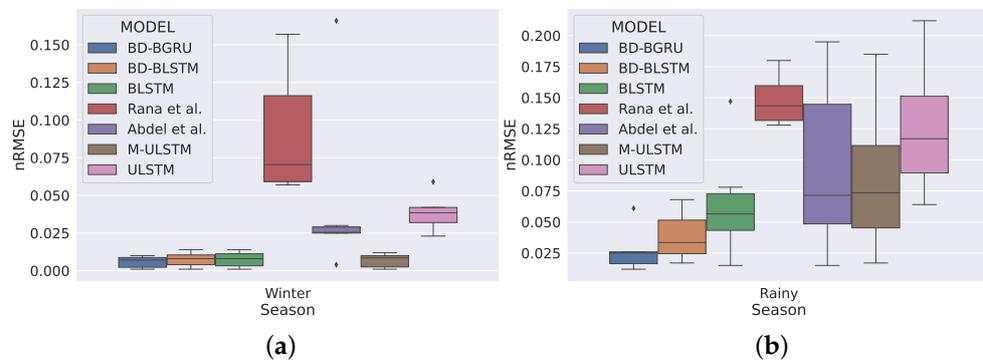


Figure 7. Season specific variability in predictions.

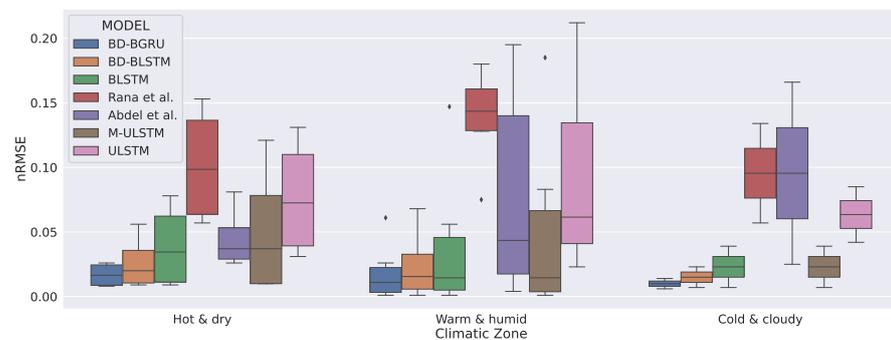


Figure 8. Climatic-zone specific variability in predictions.

Table 10 depicts the overall summary of this research work. The overall model-specific forecasting accuracy (average nRMSE) is presented individually for each station, and also model-specific standard deviation and mean rank are shown.

- We observed that BD-BGRU achieved the lowest overall nRMSE score compared to all the other models. In terms of overall forecasting accuracy, BD-BGRU dominated BD-BLSTM by 30.43%.
- BD-BGRU achieved the lowest standard deviation as 0.016. It implies that in the case of BD-BGRU the variance in predictions of GHI is minimum.
- BD-BGRU has also achieved the lowest mean rank as 1.16 compared to the other models.

Table 10. Overall performance summary of the forecasting models.

Models	Overall nRMSE	Standard Deviation (σ)	Mean Rank
[29]	0.119	0.043	6.58
[8]	0.069	0.067	4.87
ULSTM	0.082	0.058	5.91
M-ULSTM	0.045	0.038	3.58
BLSTM	0.036	0.025	3.12
BD-BLSTM	0.023	0.021	2.75
BD-BGRU *	0.016	0.016	1.16

6. Conclusions

In this paper, to forecast inter-day solar irradiation (GHI), five models namely ULSTM, M-ULSTM, BLSTM, BD-BLSTM, and BD-BGRU were implemented. A new feature

representation scheme suited for the bidirectional model was proposed. To validate the model, three different climatic zones of India, and two specific seasons, namely the winter and rainy seasons were considered. Each model was implemented for twelve city–month combinations for India. The effectiveness was established through an extensive empirical study over traditional bidirectional models and some recent state-of-the-art models.

M-ULSTM over ULSTM: Individually for each solar station, the forecasting performance of M-ULSTM was compared with ULSTM for both winter and rainy seasons. We observed that, in terms of the nRMSE, for both winter and rainy seasons, M-ULSTM outperformed ULSTM by 83.28% and 38.87%. The standard deviation was lower in the case of M-ULSTM compared to ULSTM. Hence, the above discussion suggests that, compared to ULSTM, the augmented features help M-ULSTM to predict solar irradiation more accurately.

BLSTM over M-ULSTM: We noticed that, overall, for all city–month combinations in the rainy season when the variability of GHI is observed to be higher than winter, BLSTM outperformed M-ULSTM by 59.53%. Hence, the traditional bidirectional LSTM (BLSTM) has forecasted GHI more accurately in the rainy season compared to M-ULSTM.

BD-BGRU and BD-BLSTM over BLSTM: BD-BLSTM and BD-BGRU demonstrated more generalized prediction performance compared to BLSTM. Both BD-BLSTM and BD-BGRU achieved 12.43% and 37.47% lower nRMSE scores compared to BLSTM.

BD-BGRU over BD-BLSTM: In this paper, along with bidirectional LSTM, we also implemented bidirectional GRU, and their forecasting performance was also critically compared. As per the literature, it appears that bidirectional GRU was not used earlier in this domain. The experimental results reveal, for eleven out of twelve city–month combinations, BD-BGRU outperformed BD-BLSTM by 26.67%. Hence, in Indian climatic conditions, BD-BGRU would be the definite choice for intra-day GHI forecasting.

Prediction performance of BD-BLSTM and BD-BGRU over benchmark models: In terms of mean rank, BD-BLSTM and BD-BGRU are the best two methods as compared to the benchmark models. It may be noted that bidirectional GRU may not have been used earlier for prediction in the energy domain.

The limitations of the article are as follows: (1) Including other meteorological parameters, such as temperature, humidity, cloud cover, wind speed, etc., will improve the forecasting accuracy more. However, this study was conducted on one meteorological variable, namely GHI. (2) Currently, one-year data was used at various sites for the proposed algorithm.

As future work, this study can be extended by including more input variables, more years, and including solar stations from other climatic zones.

Author Contributions: Conceptualization, S.M. and S.G.; methodology, S.M.; software, S.M.; validation, S.M., S.G., B.G., A.C., S.S.R., K.B. and A.G.R.; formal analysis, S.M.; investigation, S.M.; resources, S.M.; data curation, S.M., K.B. and A.G.R.; writing—original draft preparation, S.M.; writing—review and editing, S.M., S.G., B.G. and A.C.; visualization, S.M.; supervision, S.G., B.G. and A.C.; project administration, S.G., B.G. and A.C. All authors have read and agreed to the published version of the manuscript.

Funding: Not applicable.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Data is contained within the article.

Acknowledgments: This article resulted from the Indo-USA collaborative project, LISA 2020 between the University of Calcutta, India, and the University of Colorado, USA, and the research is supported by the National Institute of Wind Energy (NIWE) and Technical Education Quality Improvement Programme (TEQIP).

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Prāvālie, R.; Patriche, C.; Bandoc, G. Spatial assessment of solar energy potential at global scale. A geographical approach. *J. Clean. Prod.* **2019**, *209*, 692–721. doi: 10.1016/j.jclepro.2018.10.239. [[CrossRef](#)]
2. Dudley, B. BP statistical review of world energy. *BP Stat. Rev. Lond. UK* **2018**, *6*, 00116.
3. Fliess, M.; Join, C.; Voyant, C. Prediction bands for solar energy: New short-term time series forecasting techniques. *Sol. Energy* **2018**, *166*, 519–528. doi: 10.1016/j.solener.2018.03.049. [[CrossRef](#)]
4. Pan, C.; Tan, J.; Feng, D. Prediction intervals estimation of solar generation based on gated recurrent unit and kernel density estimation. *Neurocomputing* **2020**, *453*, 552–562. doi: 10.1016/j.neucom.2020.10.027. [[CrossRef](#)]
5. Alzahrani, A.; Shamsi, P.; Dagli, C.; Ferdowsi, M. Solar irradiance forecasting using deep neural networks. *Procedia Comput. Sci.* **2017**, *114*, 304–313. doi: 10.1016/j.procs.2017.09.045. [[CrossRef](#)]
6. Hejase, H.A.; Al-Shamisi, M.H.; Assi, A.H. Modeling of global horizontal irradiance in the United Arab Emirates with artificial neural networks. *Energy* **2014**, *77*, 542–552. doi: 10.1016/j.energy.2014.09.064. [[CrossRef](#)]
7. Lai, C.S.; Zhong, C.; Pan, K.; Ng, W.W.; Lai, L.L. A deep learning based hybrid method for hourly solar radiation forecasting. *Expert Syst. Appl.* **2021**, *177*, 114941. doi: 10.1016/j.eswa.2021.114941. [[CrossRef](#)]
8. Abdel-Nasser, M.; Mahmoud, K. Accurate photovoltaic power forecasting models using deep LSTM-RNN. *Neural Comput. Appl.* **2019**, *31*, 2727–2740. doi: 10.1007/s00521-017-3225-z. [[CrossRef](#)]
9. Qing, X.; Niu, Y. Hourly day-ahead solar irradiance prediction using weather forecasts by LSTM. *Energy* **2018**, *148*, 461–468. doi: 10.1016/j.energy.2018.01.177. [[CrossRef](#)]
10. Alfadda, A.; Rahman, S.; Pipattanasomporn, M. Solar irradiance forecast using aerosols measurements: A data driven approach. *Sol. Energy* **2018**, *170*, 924–939. doi: 10.1016/j.solener.2018.05.089. [[CrossRef](#)]
11. Srivastava, S.; Lessmann, S. A comparative study of LSTM neural networks in forecasting day-ahead global horizontal irradiance with satellite data. *Sol. Energy* **2018**, *162*, 232–247. doi: 10.1016/j.solener.2018.01.005. [[CrossRef](#)]
12. Cannizzaro, D.; Aliberti, A.; Bottaccioli, L.; Macii, E.; Acquaviva, A.; Patti, E. Solar radiation forecasting based on convolutional neural network and ensemble learning. *Expert Syst. Appl.* **2021**, *181*, 115167. doi: 10.1016/j.eswa.2021.115167. [[CrossRef](#)]
13. Perez, R.; Lorenz, E.; Pelland, S.; Beauharnois, M.; Van Knowe, G.; Hemker, K., Jr.; Heinemann, D.; Remund, J.; Müller, S.C.; Traummüller, W.; et al. Comparison of numerical weather prediction solar irradiance forecasts in the US, Canada and Europe. *Sol. Energy* **2013**, *94*, 305–326. doi: 10.1016/j.solener.2013.05.005. [[CrossRef](#)]
14. Voyant, C.; Notton, G.; Kalogirou, S.; Nivet, M.L.; Paoli, C.; Motte, F.; Fouilloy, A. Machine learning methods for solar radiation forecasting: A review. *Renew. Energy* **2017**, *105*, 569–582. doi: 10.1016/j.renene.2016.12.095. [[CrossRef](#)]
15. Mellit, A.; Mekki, H.; Messai, A.; Kalogirou, S.A. FPGA-based implementation of intelligent predictor for global solar irradiation, Part I: Theory and simulation. *Expert Syst. Appl.* **2011**, *38*, 2668–2685. doi: 10.1016/j.eswa.2010.08.057. [[CrossRef](#)]
16. 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*, 123285. doi: 10.1016/j.jclepro.2020.123285. [[CrossRef](#)]
17. Wan, H.; Guo, S.; Yin, K.; Liang, X.; Lin, Y. CTS-LSTM: LSTM-based neural networks for correlated time series prediction. *Knowl.-Based Syst.* **2020**, *191*, 105239. doi: 10.1016/j.knosys.2019.105239. [[CrossRef](#)]
18. Li, Y.; Zhu, Z.; Kong, D.; Han, H.; Zhao, Y. EA-LSTM: Evolutionary attention-based LSTM for time series prediction. *Knowl.-Based Syst.* **2019**, *181*, 104785. doi: 10.1016/j.knosys.2019.05.028. [[CrossRef](#)]
19. Wang, H.; Yang, Z.; Yu, Q.; Hong, T.; Lin, X. Online reliability time series prediction via convolutional neural network and long short term memory for service-oriented systems. *Knowl.-Based Syst.* **2018**, *159*, 132–147. doi: 10.1016/j.knosys.2018.07.006. [[CrossRef](#)]
20. Long, W.; Lu, Z.; Cui, L. Deep learning-based feature engineering for stock price movement prediction. *Knowl.-Based Syst.* **2019**, *164*, 163–173. doi: 10.1016/j.knosys.2018.10.034. [[CrossRef](#)]
21. Iwana, B.K.; Frinken, V.; Uchida, S. DTW-NN: A novel neural network for time series recognition using dynamic alignment between inputs and weights. *Knowl.-Based Syst.* **2020**, *188*, 104971. doi: 10.1016/j.knosys.2019.104971. [[CrossRef](#)]
22. LeCun, Y.; Bengio, Y.; Hinton, G. Deep learning. *Nature* **2015**, *521*, 436–444. [[CrossRef](#)] [[PubMed](#)]
23. Aslam, M.; Lee, J.M.; Kim, H.S.; Lee, S.J.; Hong, S. Deep Learning Models for Long-Term Solar Radiation Forecasting Considering Microgrid Installation: A Comparative Study. *Energies* **2019**, *13*, 1–15. doi: 10.3390/en13010147. [[CrossRef](#)]
24. Mukherjee, A.; Ain, A.; Dasgupta, P. Solar Irradiance Prediction from Historical Trends Using Deep Neural Networks. In Proceedings of the 2018 IEEE International Conference on Smart Energy Grid Engineering (SEGE), Oshawa, ON, Canada, 12–15 August 2018; pp. 356–361. doi: 10.1109/SEGE.2018.8499394. [[CrossRef](#)]
25. Castangia, M.; Aliberti, A.; Bottaccioli, L.; Macii, E.; Patti, E. A compound of feature selection techniques to improve solar radiation forecasting. *Expert Syst. Appl.* **2021**, *178*, 114979. doi: 10.1016/j.eswa.2021.114979. [[CrossRef](#)]
26. Feng, C.; Zhang, J. SolarNet: A sky image-based deep convolutional neural network for intra-hour solar forecasting. *Sol. Energy* **2020**, *204*, 71–78. doi: 10.1016/j.solener.2020.03.083. [[CrossRef](#)]
27. Sharadga, H.; Hajimirza, S.; Balog, R.S. Time series forecasting of solar power generation for large-scale photovoltaic plants. *Renew. Energy* **2019**. doi: 10.1016/j.renene.2019.12.131. [[CrossRef](#)]
28. Zheng, J.; Zhang, H.; Dai, Y.; Wang, B.; Zheng, T.; Liao, Q.; Liang, Y.; Zhang, F.; Song, X. Time series prediction for output of multi-region solar power plants. *Appl. Energy* **2020**, *257*, 114001. doi: 10.1016/j.apenergy.2019.114001. [[CrossRef](#)]
29. Rana, M.; Rahman, A. Multiple steps ahead solar photovoltaic power forecasting based on univariate machine-learning models and data re-sampling. *Sustain. Energy Grids Netw.* **2020**, *21*, 100286. doi: 10.1016/j.segan.2019.100286. [[CrossRef](#)]

30. Benali, L.; Notton, G.; Fouilloy, A.; Voyant, C.; Dizene, R. Solar radiation forecasting using artificial neural network and random forest methods: Application to normal beam, horizontal diffuse and global components. *Renew. Energy* **2019**, *132*, 871–884. doi: 10.1016/j.renene.2018.08.044. [CrossRef]
31. Yang, Z.; Mourshed, M.; Liu, K.; Xu, X.; Feng, S. A novel competitive swarm optimized RBF neural network model for short-term solar power generation forecasting. *Neurocomputing* **2020**, *397*, 415–421. doi: 10.1016/j.neucom.2019.09.110. [CrossRef]
32. Torres-Barrán, A.; Alonso, Á.; Dorransoro, J.R. Regression tree ensembles for wind energy and solar radiation prediction. *Neurocomputing* **2019**, *326*, 151–160. doi: 10.1016/j.neucom.2017.05.104. [CrossRef]
33. Wang, K.; Qi, X.; Liu, H. Photovoltaic power forecasting based LSTM-Convolutional Network. *Energy* **2019**, *189*, 116225. doi: 10.1016/j.energy.2019.116225. [CrossRef]
34. Malakar, S.; Goswami, S.; Ganguli, B.; Chakrabarti, A.; Roy, S.S.; Boopathi, K.; Rangaraj, A. Designing a long short-term network for short-term forecasting of global horizontal irradiance. *SN Appl. Sci.* **2021**, *3*, 477. doi: 10.1007/s42452-021-04421-x. [CrossRef]
35. Wang, K.; Li, K.; Zhou, L.; Hu, Y.; Cheng, Z.; Liu, J.; Chen, C. Multiple convolutional neural networks for multivariate time series prediction. *Neurocomputing* **2019**, *360*, 107–119. doi: 10.1016/j.neucom.2019.05.023. [CrossRef]
36. Ghimire, S.; Deo, R.C.; Raj, N.; Mi, J. Deep solar radiation forecasting with convolutional neural network and long short-term memory network algorithms. *Appl. Energy* **2019**, *253*, 113541. doi: 10.1016/j.apenergy.2019.113541. [CrossRef]
37. Hochreiter, S.; Schmidhuber, J. Long short-term memory. *Neural Comput.* **1997**, *9*, 1735–1780. doi: 10.1162/neco.1997.9.8.1735. [CrossRef] [PubMed]
38. Hochreiter, S. The vanishing gradient problem during learning recurrent neural nets and problem solutions. *Int. J. Uncertain. Fuzziness Knowl. Based Syst.* **1998**, *6*, 107–116. doi: 10.1142/S0218488598000094. [CrossRef]
39. Pascanu, R.; Mikolov, T.; Bengio, Y. Understanding the exploding gradient problem. *arXiv* **2012**, arXiv:1211.5063.
40. Chung, J.; Gulcehre, C.; Cho, K.; Bengio, Y. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv* **2014**, arXiv:1412.3555.
41. Cho, K.; Van Merriënboer, B.; Bahdanau, D.; Bengio, Y. On the properties of neural machine translation: Encoder-decoder approaches. *arXiv* **2014**, arXiv:1409.1259.
42. Wichrowska, O.; Maheswaranathan, N.; Hoffman, M.W.; Colmenarejo, S.G.; Denil, M.; Freitas, N.; Sohl-Dickstein, J. Learned optimizers that scale and generalize. In Proceedings of the International Conference on Machine Learning, Sydney, Australia, 6–11 August 2017; pp. 3751–3760.
43. Zhang, Z. Improved adam optimizer for deep neural networks. In Proceedings of the 2018 IEEE/ACM 26th International Symposium on Quality of Service (IWQoS), Banff, AB, Canada, 4–6 June 2018; pp. 1–2.
44. Bottou, L. Stochastic gradient descent tricks. In *Neural Networks: Tricks of the Trade*; Springer: Berlin/Heidelberg, Germany, 2012; pp. 421–436.
45. Graves, A.; Schmidhuber, J. Framewise phoneme classification with bidirectional LSTM and other neural network architectures. *Neural Netw.* **2005**, *18*, 602–610. doi: 10.1016/j.neunet.2005.06.042. [CrossRef]
46. Graves, A.; Mohamed, A.R.; Hinton, G. Speech recognition with deep recurrent neural networks. In Proceedings of the 2013 IEEE International Conference on Acoustics, Speech and Signal Processing, Vancouver, BC, Canada, 26–31 May 2013; pp. 6645–6649. doi: 10.1109/ICASSP.2013.6638947. [CrossRef]
47. Iqbal, M. *An Introduction to Solar Radiation*; Elsevier: Amsterdam, The Netherlands, 2012.
48. Géron, A. *Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems*; O'Reilly Media: Newton, MA, USA, 2019.
49. Wazwaz, A.M. The tanh method for traveling wave solutions of nonlinear equations. *Appl. Math. Comput.* **2004**, *154*, 713–723. doi: 10.1016/S0096-3003(03)00745-8. [CrossRef]
50. Kingma, D.P.; Ba, J. Adam: A method for stochastic optimization. *arXiv* **2014**, arXiv:1412.6980.
51. Yoo, Y. Hyperparameter optimization of deep neural network using univariate dynamic encoding algorithm for searches. *Knowl.-Based Syst.* **2019**, *178*, 74–83. doi: 10.1016/j.knosys.2019.04.01. [CrossRef]
52. Snoek, J.; Larochelle, H.; Adams, R.P. Practical Bayesian Optimization of Machine Learning Algorithms. *Advances in Neural Information Processing Systems*. 2012; pp. 2951–2959. Available online: <https://proceedings.neurips.cc/paper/2012/file/05311655a15b75fab86956663e1819cd-Paper.pdf> (accessed on 20 November 2021).
53. Bergstra, J.S.; Bardenet, R.; Bengio, Y.; Kégl, B. Algorithms for Hyper-Parameter Optimization. *Advances in Neural Information Processing Systems*. 2011; pp. 2546–2554. Available online: <https://hal.inria.fr/hal-00642998> (accessed on 20 November 2021).
54. Bergstra, J.; Yamins, D.; Cox, D.D. Hyperopt: A python library for optimizing the hyperparameters of machine learning algorithms. In *Proceedings of the 12th Python in Science Conference*; Citeseer: University Park, PA, USA, 2013; pp. 13–20.
55. Pedregosa, F.; Varoquaux, G.; Gramfort, A.; Michel, V.; Thirion, B.; Grisel, O.; Blondel, M.; Prettenhofer, P.; Weiss, R.; Dubourg, V.; et al. Scikit-learn: Machine learning in Python. *J. Mach. Learn. Res.* **2011**, *12*, 2825–2830.
56. Bergstra, J.; Bengio, Y. Random search for hyper-parameter optimization. *J. Mach. Learn. Res.* **2012**, *13*, 281–305.