Research Article

Designing a long short-term network for short-term forecasting of global horizontal irradiance

Sourav Malakar[1] · Saptarsi Goswami[2] · Bhaswati Ganguli[1] · Amlan Chakrabarti[1] · Sugata Sen Roy[1] · K. Boopathi[3] · A. G. Rangaraj[3]

Received: 25 November 2019 / Accepted: 23 February 2021 / Published online: 20 March 2021

Abstract

Long short-term memory (LSTM) models based on specialized deep neural network-based architecture have emerged as an important model for forecasting time-series. However, the literature does not provide clear guidelines for design choices, which affect forecasting performance. Such choices include the need for pre-processing techniques such as deseasonalization, ordering of the input data, network size, batch size, and forecasting horizon. We detail this in the context of short-term forecasting of global horizontal irradiance, an accepted proxy for solar energy. Particularly, short-term forecasting is critical because the cloud conditions change at a sub-hourly having large impacts on incident solar radiation. We conduct an empirical investigation based on data from three solar stations from two climatic zones of India over two seasons. From an application perspective, it may be noted that despite the thrust given to solar energy generation in India, the literature contains few instances of robust studies across climatic zones and seasons. The model thus obtained subsequently outperformed three recent benchmark methods based on random forest, recurrent neural network, and LSTM, respectively, in terms of forecasting accuracy. Our findings underscore the importance of considering the temporal order of the data, lack of any discernible benefit from data pre-processing, the effect of making the LSTM model stateful. It is also found that the number of nodes in an LSTM network, as well as batch size, is influenced by the variability of the input data.

Keywords Solar forecasting · Global horizontal irradiance · Time series · LSTM

1 Introduction

Solar radiation is one of the most important components of alternative sources of energy [32, 37]. Accurate prediction of solar radiation is essential for several tasks like planning power generation, matching peak demand, estimating surplus or even to make purchases [12]. Generation of solar power has significant variability because of its strong dependence on atmospheric conditions [31, 33, 47]. In the context of India, energy demand has been continuously on the rise because of the rapid development and expansion of urban areas. India is among the top five counties in terms of solar energy potential with the availability of sufficient solar hot-spots. Hence, research on solar energy is quite critical for India [40].

Most solar energy forecasting has been done using Numerical Weather Prediction (NWP) [4] models, also referred to as physical models in the literature. Statistical models like Auto-Regressive Integrated Moving Average (ARIMA), Generalized Autoregressive Conditional Heteroskedasticity (GARCH), etc., and machine learning models

✉ Sourav Malakar, sourav.xaviers@gmail.com | [1]University of Calcutta, Kolkata, India. [2]Bangabasi Morning College, University of Calcutta, Kolkata, India. [3]National Institute Of Wind Energy (NIWE), Under the Ministry of New and Renewable Energy, Government of India, Chennai, India.

SN Applied Sciences (2021) 3:477 | https://doi.org/10.1007/s42452-021-04421-x

like Support Vector Regression (SVR), Artificial Neural Network (ANN), etc., have also been used for prediction.

Currently, machine learning models have emerged as state-of-the-art solar forecasting models for one to few hours ahead of forecasting [47]. Presently, many of the studies report superior performance exhibited by deep learning models as compared to machine learning models for classification, regression, and time series forecasting [3]. As noted by LeCun et al. [25] in their seminal paper, that deep learning models were doing better than machine learning models in many application domains because of their superior capability of learning complex patterns from the raw data. LSTM is a deep learning-based model specially designed to handle sequence data. Some of the advantages of the LSTM are it can handle nonlinearity in data well [24], and it can memorize long temporal relationships in data, etc. Over the years, LSTM-based models have shown their efficacy across various application domains like language models [9, 46], speech [49], weather forecasting [45], traffic forecasting [52], etc.

While LSTM models are considered state-of-the-art for forecasting in diverse application domains like anomaly detection [10], text classification [6], malware classification [20], the issue of representing the time-series does not seem to have a clear approach. Either the time-series data is represented in a (a) Supervised setup, where the previous time steps are considered to be independent of each other and are treated as separate features, or in a (b) Non-Supervised setup, where the ordering and time steps are given importance. Suppose we have a time-series of length ***n*** given by X_1, X_2, X_3,..., X_n. A sequence, X_1, X_2, X_3,..., X_n is converted into the following representation, {[X_1, X_2, X_3, X_4] [X_5]}, {[X_2, X_3, X_4, X_5] [X_6]}, ..., {[X_{n-4}, X_{n-3}, X_{n-2}, X_{n-1}] [X_n]}, assuming a window of size four. The observations are separated by a comma and are enclosed by curly braces. Each observation consists of two parts, the input features and the output both enclosed by square brackets. It is to be noted, rather than using all the observations in a single go, it can be broken into windows as shown earlier. While finding the parameters of the network often rather than using all the windows, a subset of the windows is used. These subsets are called batches. In a Non-Supervised setup, there is a choice to maintain the temporal order in three ways a) Within the same window, b) Within the batch, and c) Between batches. In paper, [14] and [50], the authors have treated the input features as independent of time.

Another important design issue is data pre-processing such as identification and removal of trend and seasonality. It is observed that some authors have made their data stationary before any model fitting [3, 13, 44], whereas some have not pre-processed the data [38].

Hence, there is a general disagreement between design choices for an LSTM, such as preserving the temporal order of the data and the need for pre-processing. Apart from the above-mentioned two design issues, it is also perceived that a few other issues like the batch size, the prediction horizon, adjustments for inherent input data variability can impact model performance. In this paper, we consider the design questions enlisted below.

- Whether LSTM benefits from pre-processing steps such as seasonality removal.
- Whether to set up the problem as a Supervised or Non-Supervised problem and in the latter case, whether it is necessary to consider dependency among batches.
- How does the forecasting performance of an LSTM change with the increase in the prediction horizon, especially in context to the season?
- What is the effect of variability in input data on model complexity?

In this paper, we have attempted to investigate the above-listed questions systematically in the context of short-term intra-day forecasting of GHI using LSTM. The major contributions of this paper are enlisted as follows.

- An empirical study has been conducted for three solar stations, two seasons, and two climatic zones in India. It may be noted that such a study is quite rare in India, despite its rich solar energy potential.
- The design questions enlisted have been empirically evaluated, and important recommendations like considering the temporal order of the data (Non-Supervised setup), no pre-processing, and preserving dependency between batches have been made.
- It has been established that the forecasting performance is dependent on batch size and variability of the input data.
- It has also been demonstrated that the number of nodes required by the LSTM network increases with an increase in the variability of the input data.
- The model obtained using these recommendations produces superior forecasting performance applying RF, RNN, and LSTM, respectively.

The rest of the paper is organized as follows. In Sect. 2, we have performed a detailed literature review of machine learning and deep learning for solar energy forecasting. The research efforts have been also categorized in terms of the length of the forecasting horizon. In Sect. 3, we have provided a brief outline of the LSTM architecture. In Sect. 4, we have discussed the materials and methods employed in setting up the experiment. In Sect. 5, the results of the forecasting models are presented along with a critical analysis. The paper concludes with a discussion in Sect. 6.

2 Related work

In this section, a brief overview of current research for the prediction of GHI has been presented. It may be noted that the approach of building physical models for estimating GHI using classical equations [27] has not been included in the scope of the furnished review. There is also a conscious effort to include studies conducted in India. In Fig. 1, the research papers or articles have been categorized in terms of the type of the forecasting model, prediction on different lengths of forecast horizons, and the number of input variables of the model.

2.1 Statistical and machine learning models

Yang et al. [51] have analyzed three approaches for one hour ahead solar irradiation forecasting based on the exponential smoothing technique (ETS) applied to cloud cover. Kashyap et al. [21] have proposed a model based on ANN to forecast GHI for a one-hour horizon. Feng et al. [11] have developed a one hour ahead GHI forecasting model using SVM classifier, with 9.75% nRMSE. Reikard et al. [42] have used their forecasting model for several horizons ranging from 15 minutes to three hours. They have reported that for a 15-minute horizon, the persistence model and regression model outperformed the frequency domain model. For a 45 minute horizon, the performance of the above three models was close. At the one-hour horizon, ARIMA achieved better accuracy when applied to the Clear Sky Index. For a two-hour horizon, the frequency domain model performed better than others. Finally, for a three-hour horizon, the performances of the frequency-domain approach and ARIMA were similar. Alfadda et al. [2] have shown that a multi-layer perceptron (MLP) works better as compared to SVR, k-nearest neighbors (kNN), and decision tree regression for one hour ahead irradiation forecasting. Fouilloy et al. [13] have proposed a bagged regression tree and RF-based models to predict hourly GHI for 6 hours. Perveen et al. [35] have proposed an adaptive neural fuzzy inference system (ANFIS)-based multivariate solar power forecasting model for different sky conditions for India. Benali et al. [5] demonstrated that RF performs best for predicting GHI, beam normal irradiation (BNI), and diffuse horizontal irradiation (DHI) for six hours ahead. Perveen et al. [34] have designed an ANFIS-based multivariate short-term solar power forecasting model for complex climatic conditions for India. Rana et al. [39] have designed five minutes to three hours ahead univariate solar photovoltaic power forecasting model based on a unique re-sampling technique and have combined multiple RF models predictions for individual steps to design a single robust multi-steps ahead prediction model.

- It can be observed that in most of the studies [2, 5, 11, 13, 21, 34, 39, 42], the authors have reported short-term forecasting models.
- Both univariate models [5, 13, 39, 42, 51], and multivariate models [2, 21, 34, 35] have been used for solar energy forecasting.
- Up to 2015, the use of statistical-based approaches [42, 51] was more.

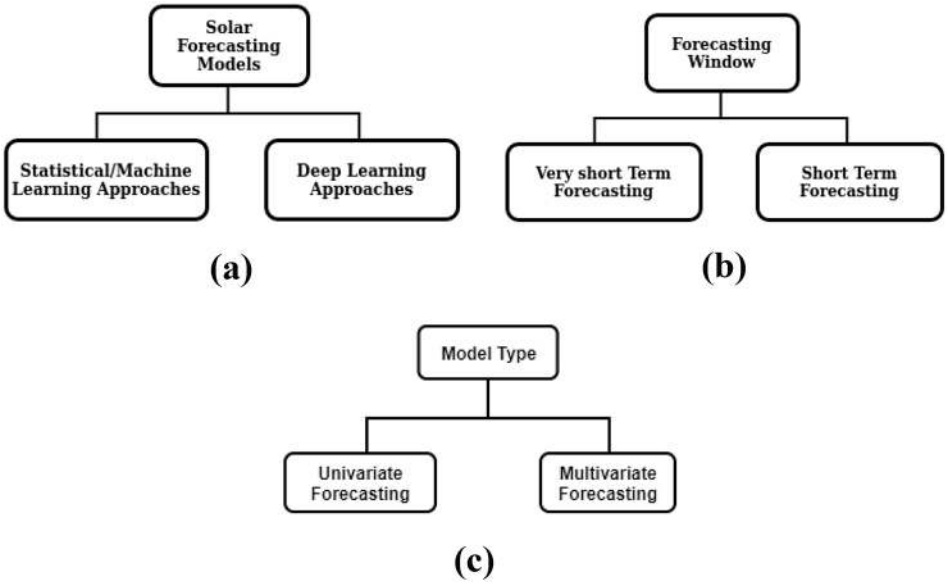


Fig. 1 Classification of solar forecasting models based on (**a**) Type of time-series forecasting model (**b**) Type of forecasting window (**c**) Number of independent variables

- Ensemble-based approaches like RF models were being deployed, and they reported better results than contemporary models [5, 13, 39].

2.2 Deep learning-based models

Ahmad et al. [3] showed the efficacy of deep recurrent neural network-based models over other benchmark models when applied to solar energy data in Canada. Qing et al. [38] have achieved better results using LSTM as compared to neural networks using backpropagation. Caballero et al. [7] have designed a forecasting model using LSTM to forecast solar irradiation for the window of five minutes. Mukherjee et al. [28] have proposed an LSTM-based multivariate solar forecasting model for Kharagpur, India. Caldas et al. [8] have designed a hybrid forecasting model by considering solar energy data and sky images to predict for one to ten minutes ahead. Nikitidou et al. [29] have designed a 15–240 mins ahead model for forecasting cloudiness. Ryu et al. [43] have reported a forecasting model to forecast 5 to 20 mins ahead of using a convolutional neural network (CNN)-based model on total sky images and lagged value of GHI. Abdel et al. [1] have proposed a univariate photovoltaic power forecasting model for hourly data based on an LSTM-RNN while experimenting with five different model architectures. Li et al. [26] have reported that RNN-based solar power forecasting model has outperformed the persistence method, backpropagation neural network (BPNN), radial basis function (RBF) neural network, SVM, and LSTM. Huang et al. [18] have proposed an hourly LSTM-MLP-based GHI forecasting model. Kumari et al. [23] have designed an hourly GHI forecasting model using an ensemble approach. The extreme gradient boosting forest (XGBF) and deep neural networks (DNN) used as the base learners. Ridge regression is used to combine the predictions.

- It can be observed that the research has been conducted for both short-term forecasting [7, 8, 42, 43], and very short-term forecasting [1, 11, 18, 23, 26, 29, 48].
- From 2017 to 2020, in many of the studies [1, 3, 7, 26, 29], the univariate forecasting models have been employed. It is also observed that LSTM models have been increasingly used for solar energy forecasting.

It can be observed that the literature for India is limited even though the country has rich solar potential. Most available papers [8, 18, 26, 28, 29, 43] consider prediction at a coarse time resolution or are limited to one geographical region or a particular solar power plant.

We have compared the performance of our proposed method with three recent methods for solar power forecasting. These include the RF and RNN of [39] and [26] and the LSTM developed by [1].

- In paper [39], the authors have used the same algorithm repetitively for multi-step prediction and have tuned model hyperparameters such as the number of trees and splits using Grid Search with 10-fold cross-validation.
- In paper [1], the authors have used a specific LSTM architecture for univariate solar power forecasting.
- In paper [26], the authors have reported RNN for inter and intra-day prediction.

3 Deep learning sequence model and LSTM

Feed forward neural network (FNN) is the most common type of deep learning architecture. It has demonstrated remarkable performance across application domains over traditional machine learning problems. However, one of the limitations of FNNs is their inability to handle sequence data like text, video, time-series, etc. The RNN with a memory component, where the current output is a function of current input as well as a previous step can handle this issue. Though RNN achieved reasonable success, one of the weaknesses later exposed was its inability to remember long-range dependency because of the vanishing gradient problems [16].

LSTM was proposed by Hochreiter and Schmidhuber [17] and can address vanishing and exploding gradients [16, 36]. LSTM is specially designed to memorize very long-term temporal dependencies through memory cells containing several types of gates. Apart from that, LSTM can learn nonlinearity. Hereby, using a schematic diagram in Fig. 2, the detailed architecture of a specific memory cell of LSTM is shown. The mathematical equations associated with different gates of the LSTM cell are discussed with the description of the gates.

Suppose at time *t* the current input is x_t and the previous hidden state is h_{t-1}, then the current hidden state h_t and the current cell state c_t are computed as follows:

- *Forget gate*$(f_t) = \sigma(w_f[h_{t-1}, x_t] + b_f)$: Depending on the current input x_t and previous hidden layer output h_{t-1}, based on a sigmoid layer, forget gate produces either 0 or 1. If 1, memory information is retained, else discarded.
- *Input gate*$(i_t) = \sigma(w_i[h_{t-1}, x_t] + b_i)$: Input gate helps to decide on new information to be added to the current

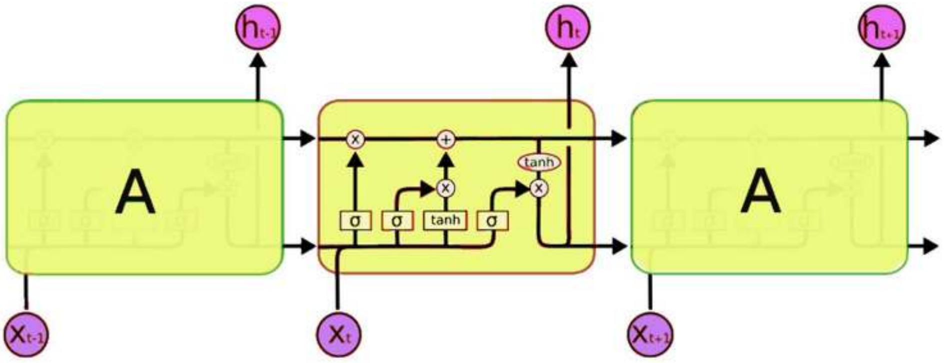


Fig. 2 https://colah.github.io/posts/2015-08-Understanding-LSTM [30] Single memory cell architecture of LSTM

cell state based on new candidate values provided by $\hat{c}_t$.

- *Cell state*$(c_t) = tanh(w_c[h_{t-1}, x_t] + b_c)$: The new cell state c_t depends on the previous cell state c_{t-1} and $c_{t-1} * f_t$ is the fraction of the old cell state that will be discarded with the help of forget gate, while new information will be added through $\hat{c}_t * i_t$. The summation of these two simultaneous updates is the current cell state.
- *Output gate*$(o_t) = \sigma(w_o[h_{t-1}, x_t] + b_o)$: Long-term state output is determined based on a sigmoid activation function.
- *Hidden state*$(h_t) = o_t * tanh(c_t)$: Finally, the result of the output gate is multiplied with the cell state through tanh to compute the value of the current hidden state.

Here, w_f, w_i, w_c, and w_o are weight matrices. b_f, b_i, b_c, and b_o are the biases for individual gates. σ indicates a sigmoid activation function. * stands for element wise multiplication, and + implies element wise addition.

- The LSTM model is trained by selecting a continuous portion or window from the input data. Instead of taking all such windows for training, often it is broken into batches.
- If the batches are considered independent of each other, then such a model is called a stateless model. While if batch to batch dependency is taken into account, then it is called a stateful model.
- Typically, when dealing with the sequence data, the hidden layer nodes are the LSTM cells. In Fig. 3, a simple schematic diagram of a deep neural network is shown, whereas a basic building block in the hidden layers, the LSTM cells are used. The inputs and the outputs are denoted as $[I_1, I_2, I_3, ..., I_n]$, and $[O_1, ..., O_n]$, respectively.
- Like traditional neural network, 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, etc.

4 Materials and methods

This section has five subsections. In the first subsection, the source of the data, the extraction process, time-period, etc., have been outlined. In the second subsection, the pre-processing steps are described briefly. It is needed to understand how the design issue of pre-processing is investigated in this paper. In the third subsection, how the

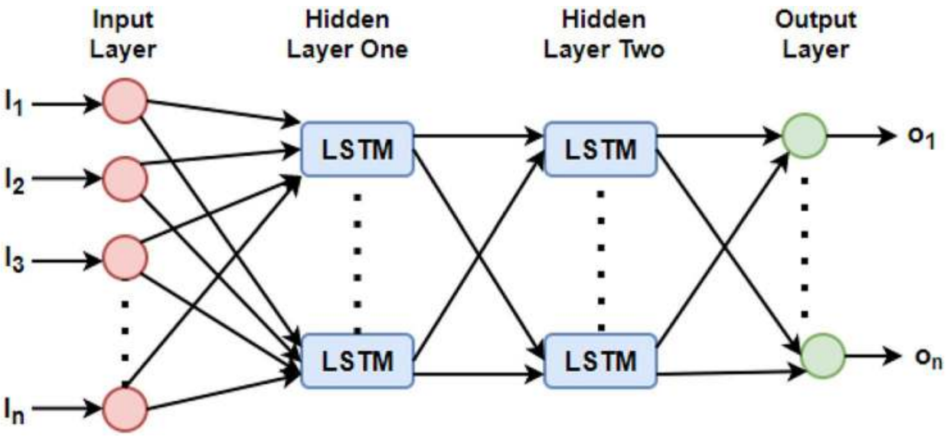


Fig. 3 A neural network based on LSTM cels

design issue of temporal order (Supervised versus Non-Supervised) is set up for the experiment is elaborated. In the fourth subsection, the proposed LSTM-based architectures are discussed in detail. Finally, in the fifth subsection, for the evaluation of the forecasting model, different error metrics are furnished.

4.1 Data collection

Indian Ministry of New and Renewable Energy (MNRE) in 2011 has initiated extensive solar and meteorological monitoring under the Solar Radiation Resource Assessment (SRRA) project [22]. Indian climate is made up of four seasons namely Summer, Monsoon (rainy) season, Post-Monsoon, and Winter by the Indian Meteorological Department (IMD).

We have used the application programming interface (API) provided by the Center for Wind Energy Technology (C-WET) to crawl raw solar irradiation data for SRRA stations across India. In this paper, data for 2016 were used for two climatic zones (Hot and Dry and Hot and Humid), and three stations located at Chennai (Tamil Nadu), Howrah (West Bengal), and Ajmer (Rajasthan). Table 1 describes the details of the solar stations, date range, number of data elements, etc. For each of the stations, we have chosen a month each from the rainy and winter seasons. Typically, the rainy season is known for its high variability in GHI compared to winter.

In Fig. 4, the distribution and variability of GHI are illustrated for each station-season combination. The plot shows that the variability of GHI is higher for Howrah and Ajmer in the rainy season, whereas it is relatively lesser for the other cases. The box-plots also confirm the absence of any outliers. The data for Howrah in the rainy season are observed to be having maximum skew compared to other stations.

Table 1 Description of the data

City	Month	Season	Period of data	Number of samples	Latitude	Longitude	Climatic zone
Chennai	December	Winter	01/12/2016 to 31/12/2016	44640	12.96N	80.22E	Hot and humid
Howrah	December	Winter	01/12/2016 to 31/12/2016	44640	22.55N	88.31E	Hot and humid
Ajmer	January	Winter	01/01/2016 to 31/01/2016	44640	26.40N	74.66E	Hot and dry
Chennai	November	Rainy	01/11/2016 to 30/11/2016	43200	12.96N	80.22E	Hot and humid
Howrah	August	Rainy	01/08/2016 to 31/08/2016	44640	22.55N	88.31E	Hot and humid
Ajmer	August	Rainy	01/08/2016 to 31/08/2016	44640	26.40N	74.66E	Hot and dry

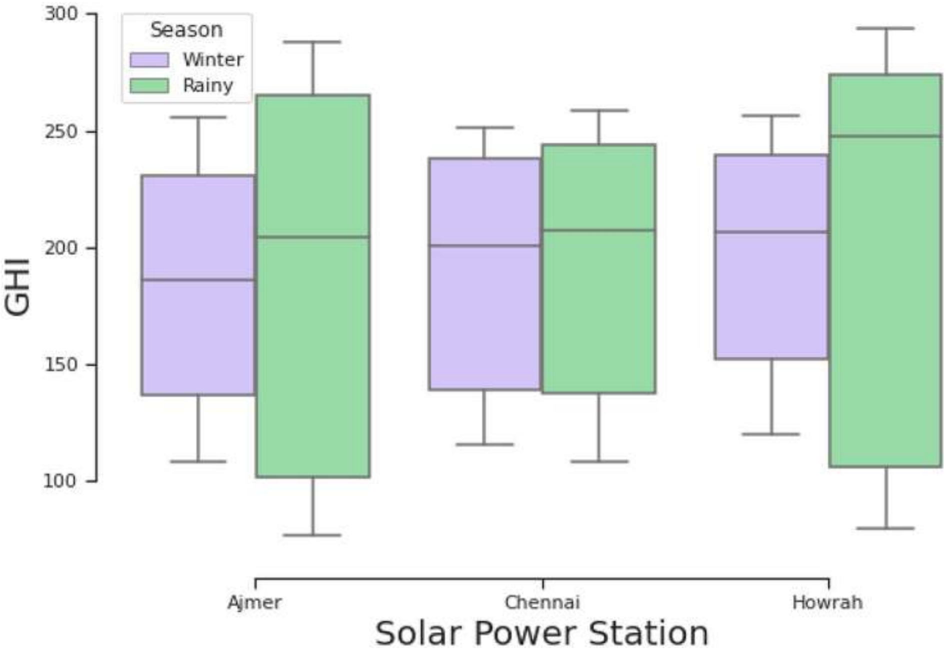


Fig. 4 Box plot of GHI (W/m^2) across of solar power station

4.2 Data pre-processing

In the pre-processing, firstly, the night hours are removed [19].

- As per standard practice for short-term solar forecasting, the resolution of GHI is converted from one minute into 5 mins [39, 41].
- For each day, we have filtered the GHI between 7 AM to 7 PM. After removing the night hours, we have combined all days in a month to construct a single time-series.
- The GHI values have been normalized to lie in [0, 1] using Eq. 1.

$$\widehat{GHI_t} = \frac{GHI_t - GHI_{min}}{GHI_{max} - GHI_{min}} \tag{1}$$

In Eq. 1, GHI_t is the GHI at time-step t, GHI_{min} is the minimum value of the population, GHI_{max} is the maximum value of the population, and $\widehat{GHI_t}$ is the normalized value of GHI at time-step t.

In some papers [44, 52], the authors have removed the non-stationary part of the series before fitting the data into a deep learning model like LSTM. Our data display daily seasonality. Hence, we have deseasonalized it using the following algorithm.

```
Result: $GHI_s$
$GHI_s$ = [ ];
interval = 144;  //12 readings per hour multiplied by 12 hours.
j = 1;
for $i$ = interval to length(Series) do
    diff = Series[i] - Series[i-interval];
    $GHI_s$[j] = diff;
    j++;
end
```

Algorithm 1: Series_to_Deseasonalize

- Thus, from the *i*th observation, we have subtracted the $(i - 144)^{th}$ observation to remove day-wise seasonality and then appended it sequentially. Here, GHI_s is the final series.
- The raw time-series and the pre-processed time-series are used as input to the LSTM network and compared.

4.3 Supervised or non-supervised learning

In this section, we have outlined the experimental setup needed to investigate the design issue of whether to preserve the temporal dependency (Non-Supervised) or not (Supervised).

For LSTM, the preparation of data is different than for traditional machine learning algorithms. The data 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 Fig. 5, the array is pictorially presented. The input features are denoted as $Feature_1$, $Feature_2$, $Feature_3$,..., and $Feature_n$. The time-steps are represented as T_1, T_2, T_3,..., and T_m.

- In the Supervised setup, the array size is taken as (72, 1, 30).
- In the Non-Supervised setup, the array size is taken as (72, 30, 1).

4.4 LSTM architectures

In this subsection, the details of LSTM networks are discussed.

- We have used a sequential model from the Keras Library [15] to design four sequentially arranged layers including, an input layer, two LSTM hidden layers, and one output layer.

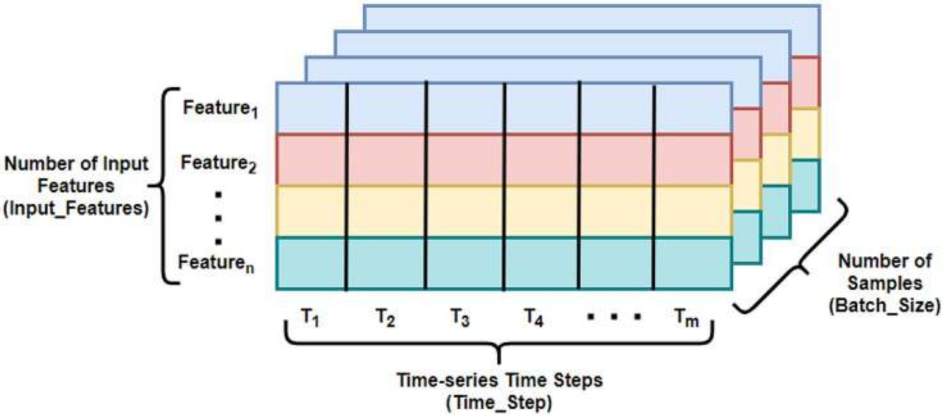


Fig. 5 A schematic diagram of a 3D array for LSTM

Table 2 Hyper-parameters to optimize

Models	Hyper-parameter	Value
	Number of hidden layers	1, 2, 3
	Nodes in hidden layers	25, 50, 75 , 100, 125, 150
DSS-LSTM/ DSSL-LSTM	Learning rate	0.1, 0.01, 0.001
	Batch size	1, 9, 18, 36, 72
	Epoch	25, 50, 100, 200

- At each LSTM layer, the weights have been initialized with random weights using a normal distribution.
- We have stored the best forecasting model using the technique Callbacks provided by Keras. The last layer is a Dense layer with 20 nodes, where a forecasting window of 1 hour 40 minutes has been used.
- The hyper-parameters settings are presented in Table 2. We have used Adam as the optimizer with a learning rate set to 0.01. Hyper-parameters like the number of epochs, number of layers, batch size, learning rate, and the number of nodes in each hidden layer has been optimized using the Random Search approach with 5-fold cross-validation with three repetitions of the process.
- Tanh activation has been used for each hidden layer.
- At the time of prediction, we have considered different specifications of a batch size such as 1, 9, 18, 36, and 72 to find the optimal choice.
- The stateful parameter is set to True and False alternatively to investigate the effect of preserving dependency between batches.

The stateful LSTM has been referred to as DSS-LSTM and stateless LSTM as DSSL-LSTM for subsequent discussions.

4.5 Evaluation of forecasting model

We have used three evaluation metrics namely root mean square error (RMSE), normalized root mean square error (nRMSE), and Explained Variance Score. The following equations are used for calculating the evaluating metrics.

$$\text{RMSE} = \sqrt{\frac{\sum_{i=1}^{n}(GHI_t - \widehat{GHI}_t)^2}{n}} \tag{2}$$

In Eq. 2, GHI_t is the tth actual value and the corresponding predicted value is $\widehat{GHI}_t$. nRMSE can be a good measure of forecasting error when we want to forecast for multiple data-sets. It is defined as follows.

Table 3 Comparison of stateful LSTM on raw and deseasonalized data

Name of the station	DSS-LSTM (Raw Data)		DSS-LSTM (Pre-processed Data)	
	nRMSE	Explained_Variance_Score (%)	nRMSE	Explained Variance Score (%)
Chennai-December	0.023	99.84	0.529	77.21
Howrah-December	0.021	99.92	0.670	75.15
Ajmer-January	0.027	99.88	0.235	97.11
Chennai-November	0.021	99.86	0.125	79.28
Howrah-August	0.021	99.83	0.078	99.81
Ajmer-August	0.022	99.82%	0.102	99.76%

$$\text{nRMSE} = \frac{RMSE}{\sigma} \tag{3}$$

$$n\text{RMSE}(\%) = \frac{\text{RMSE}}{\sigma} \times 100 \tag{4}$$

In Eqs. 3 and 4, σ is the standard deviation of the actual values of GHI. Explained Variance Score is given in the below equation.

$$\text{Explained Variance Score} = \frac{Var\{GHI - \widehat{GHI}\}}{Var\{GHI\}} \tag{5}$$

5 Results

This section has six subsections. In the first subsection, the performance of DSS-LSTM is evaluated on the raw and pre-processed time-series. In the second subsection, we have investigated whether to set up the time-series prediction problem as Supervised or Non-Supervised. In the third subsection, the effect of batch size is examined on the forecasting performance of DSS-LSTM. In the fourth subsection, the performance of DSS-LSTM is analyzed with different choices of prediction horizons. In the fifth subsection, the forecasting performance of DSS-LSTM is analyzed with the station-season specific variability of GHI. Finally, in the sixth subsection, the overall forecasting performance of DSS-LSTM is compared with the benchmark models.

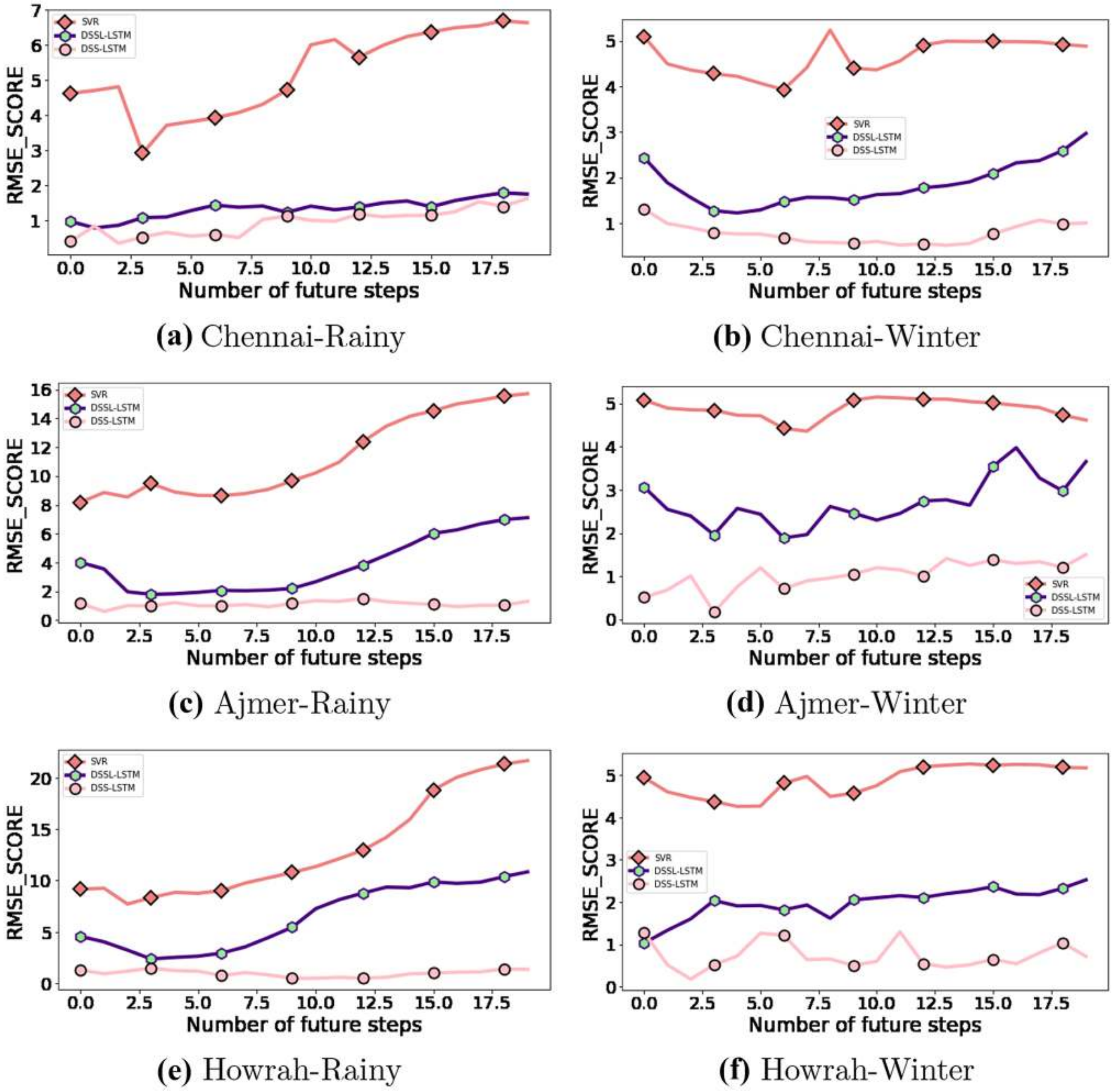
5.1 Importance of data pre-processing

Table 3 gives the overall forecasting accuracy in terms of nRMSE and the Explained Variance Score. It is observed that DSS-LSTM has a better nRMSE score if it is dealing with raw time-series. It is found that with raw data and under all climatic conditions, DSS-LSTM better captures data variability. For data-sets corresponding to Howrah-Winter, Chennai-Rainy, and Chennai-Winter, the model is capable of explaining 20-24% more variability in terms of Explained Variance Score as compared to when the data are pre-processed.

5.2 Supervised or non-supervised?

Here, we have presented the comparison between SVR (Supervised), Stateless LSTM (Non-Supervised within a batch), and Stateful LSTM (Non-Supervised across batches).

Figure 6 compares the RMSE scores for 20 steps ahead prediction of GHI. We have observed that,



(a) Chennai-Rainy

(b) Chennai-Winter

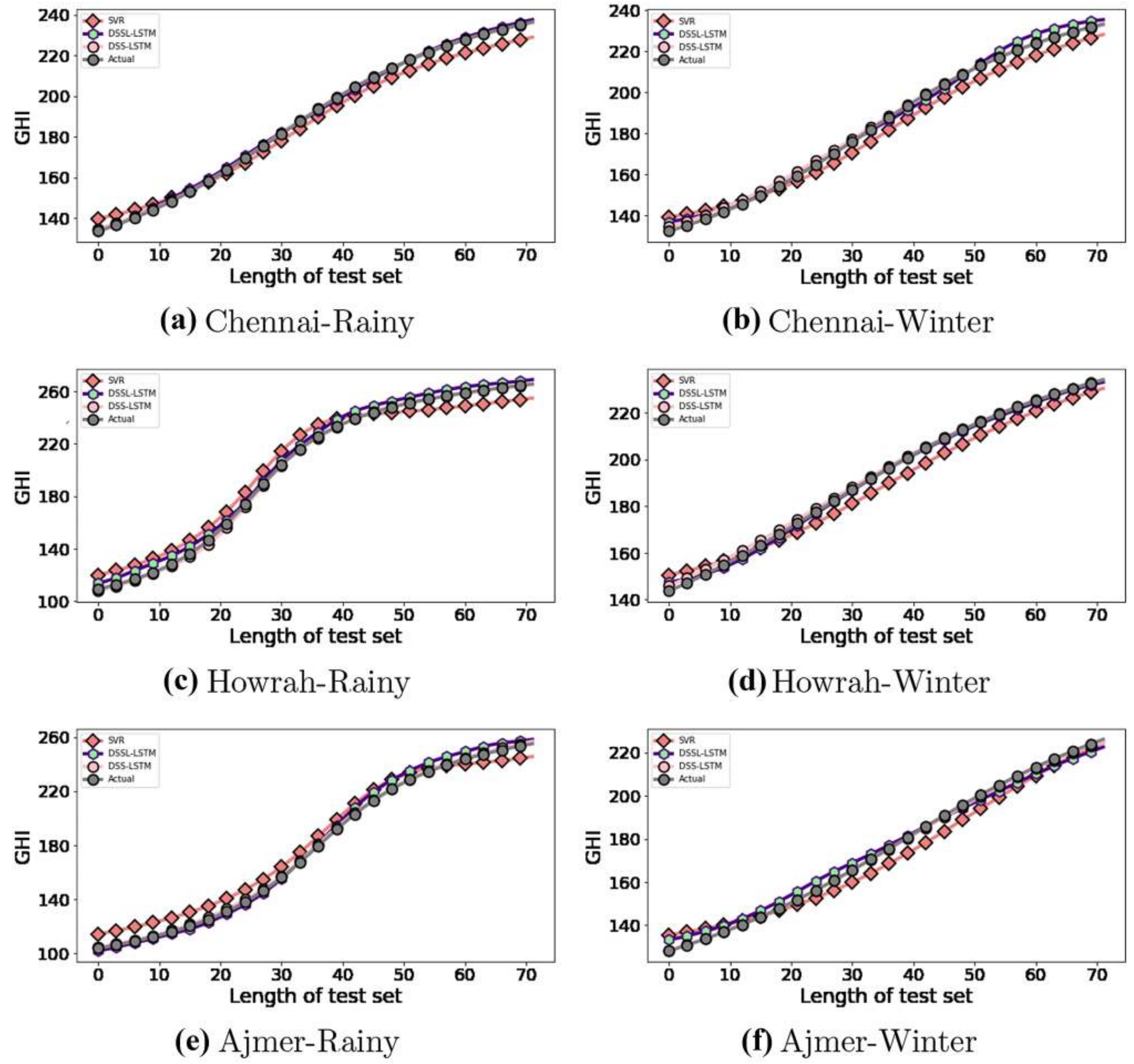
(c) Ajmer-Rainy

(d) Ajmer-Winter

(e) Howrah-Rainy

(f) Howrah-Winter

Fig. 6 RMSE for 20 steps ahead prediction of GHI (W/m^2)



(a) Chennai-Rainy

(b) Chennai-Winter

(c) Howrah-Rainy

(d) Howrah-Winter

(e) Ajmer-Rainy

(f) Ajmer-Winter

Fig. 7 Forecasting of GHI (W/m^2)

Table 4 Comparison of DSS-LSTM with DSSL-LSTM and SVR on overall nRMSE-Score

Name of the Station	Supervised	Non-Supervised	
	SVR	DSSL-LSTM	DSS-LSTM
Chennai-Winter	0.148	0.056	0.023
Howrah-Winter	0.183	0.075	0.021
Ajmer-Winter	0.155	0.088	0.027
Chennai-Rainy	0.170	0.043	0.021
Howrah-Rainy	0.289	0.149	0.021
Ajmer-Rainy	0.223	0.081	0.022

- For all climatic zones, the performance of DSS-LSTM is more stable.
- SVR produced notably higher RMSE scores.
- In the rainy season, when the variability of GHI is high, DSS-LSTM has outperformed other models.
- For both climatic zones, LSTM (stateful) outperformed LSTM (stateless).

In Table 4, we have observed that, for all climatic zones, the Non-Supervised approach outperformed the Supervised approach. Figure 7 shows forecasted GHI for the test set. For all climatic conditions, the DSS-LSTM

Table 5 Comparison of different batch size on nRMSE score

Name of the Station	Batch size				
	(1)	(9)	(18)	(36)	(72)
Chennai-Winter	0.029	0.025	0.025	0.030	0.023
Howrah-Winter	0.029	0.029	0.029	0.029	0.021
Ajmer-Winter	0.034	0.034	0.034	0.034	0.027
Chennai-Rainy	0.039	0.033	0.031	0.032	0.021
Howrah-Rainy	0.033	0.026	0.025	0.021	0.021
Ajmer-Rainy	0.033	0.026	0.025	0.022	0.022

has outperformed other models. In all the cases, SVR has produced the worst predictions.

5.3 Effect of batch size

The test set has been split using alternative batch sizes of 1, 9, 18, 36, and 72. Table 5 compares the corresponding nRMSEs. Compared to a batch size of 9, 18, and 36, a batch size of 72 has produced approximately 28.64%, 25.50%, and 24.47% better nRMSE.

As illustrated in Fig. 8,

- For Chennai-Winter, Howrah-Winter, Ajmer-Winter, and Chennai-Rainy having the lower variability of GHI, as we increase batch size, the nRMSE increases and we get the best nRMSE for a batch size of 72.
- However, in the case of Howrah-Rainy and Ajmer-Rainy having higher variability in GHI, the nRMSE increases as we increase batch size, but saturates for a batch size of 36.

Hence, the above discussion suggests that in the case of solar forecasting, for stations with high variability of GHI, smaller batch size is recommended for LSTM. However, for stations with lower variability of GHI, a bigger batch size will give better forecasting performance.

5.4 Prediction horizon

Table 6 shows nRMSE scores for alternative prediction horizons, and we have observed that the best results are obtained for 20 steps ahead prediction. The network structure of 20 steps used to forecast other forecasting horizons as well, such as 25 (2 hours 5 minutes) and 30 steps (2 hours 30 minutes). Increasing the prediction horizon from 20 to 25 and 20 to 30 increases the nRMSE by 16.72% and 31.88%, respectively. It may, however, be noted that for the rainy season during which GHI is more variable, the effect of increasing the prediction horizon on forecasting accuracy is larger.

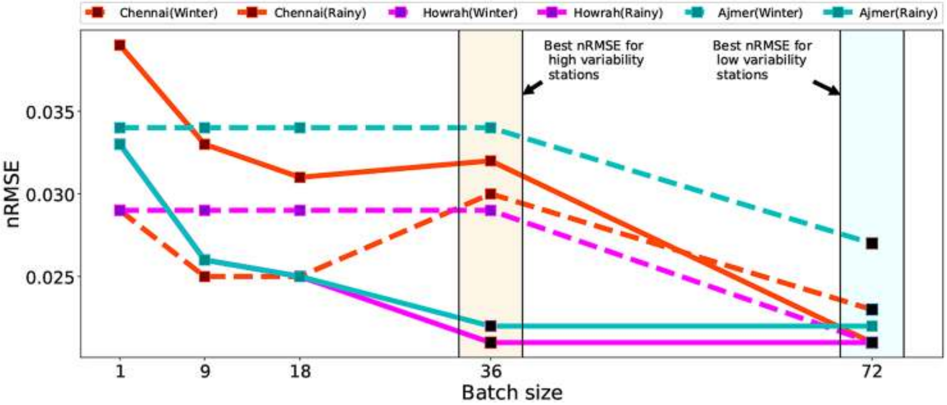


Fig. 8 Batch size is compared in terms of forecasting performance (nRMSE)

Table 6 Performance of stateful LSTM for different prediction horizons

Name of the Station	nRMSE score				
	(20 steps)	(25 steps)	(% of increase)	(30 steps)	(% of increase)
Chennai-Winter	0.023	0.026	13.04	0.027	17.39
Howrah-Winter	0.021	0.022	4.76	0.022	4.76
Ajmer-Winter	0.027	0.029	7.40	0.032	18.51
Chennai-Rainy	0.021	0.026	23.80	0.029	38.09
Howrah-Rainy	0.021	0.027	28.57	0.037	76.19
Ajmer-Rainy	0.022	0.027	22.72	0.030	36.36

5.5 Input variability vs Network complexity

Here, the complexity of the DSS-LSTM models measured in terms of the number of hidden layer nodes is analyzed in context to variability in GHI. Out of the six input conditions Howrah-Rainy and Ajmer-Rainy exhibit maximum variability in terms of GHI. To perform this analysis, we have increased the number of hidden layers from 25 to 150, with a step size of 25.

As illustrated in Table 7 and Fig. 9,

- The example cases of Chennai-Winter, Howrah-Winter, Ajmer-Winter, and Chennai-Rainy having the lower input variability need fifty nodes for optimal performance measured in terms of nRMSE.
- However, for example, cases of Howrah-Rainy and Ajmer-Rainy having higher variability in GHI need hundred nodes for giving optimal performance measured in terms of nRMSE.

This supports the existing knowledge that higher variability in solar data needs more model parameters or nodes to achieve an adequate forecasting performance.

Table 7 Network complexity of DSS-LSTM is compared against the station wise variability of GHI (W/m^2)

Name of the Station	nRMSE for different hidden nodes						Optimal nodes	Variability of GHI (W/m^2)
	25	50	75	100	125	150		
Chennai-Winter	0.028	**0.023**	0.024	0.026	0.026	0.028	50	48.26
Howrah-Winter	0.032	**0.021**	0.023	0.023	0.025	0.027	50	45.16
Ajmer-Winter	0.035	**0.027**	0.028	0.029	0.031	0.032	50	47.64
Chennai-Rainy	0.028	**0.021**	0.022	0.023	0.024	0.025	50	51.88
Howrah-Rainy	0.026	0.024	0.022	**0.021**	0.023	0.025	100	80.21
Ajmer-Rainy	0.031	0.026	0.023	**0.022**	0.022	0.024	100	78.32

The bold notations are used to depict the minimum nRMSE scores for an optimal number of nodes in the hidden layers of the network (Network complexity)

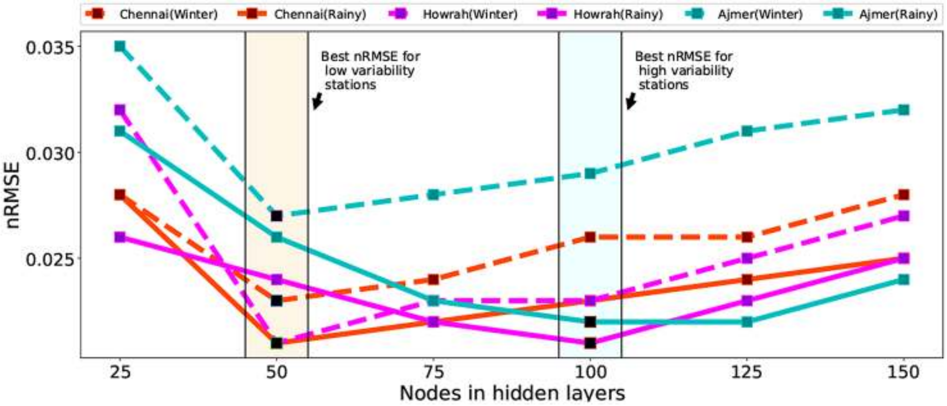


Fig. 9 Network complexity is compared in terms of forecasting performance (nRMSE)

Table 8 Comparison of DSS-LSTM with current approaches on overall nRMSE-Score (W/m^2)

Name of the station	Abdel et al. [1]	Li et al. [26]	Rana et al. [39]	DSS-LSTM*
Chennai-Winter	0.027	0.024	0.130	0.023
Howrah-Winter	0.026	0.063	0.157	0.021
Ajmer-Winter	0.057	0.068	0.057	0.027
Chennai-Rainy	0.086	0.039	0.121	0.021
Howrah-Rainy	0.102	0.198	0.166	0.021
Ajmer-Rainy	0.111	0.171	0.131	0.022
Overall nRMSE	0.068	0.099	0.127	0.022
Mean Rank	2.416	3.083	3.500	1.000

5.6 Comparison to other prediction approaches

In Table 8, the overall prediction performance of DSS-LSTM has been compared to the prediction performance of the methods suggested by Rana et al. [39], Abdel et al. [1], and Li et al. [26]. It has been observed that DSS-LSTM has produced a lower nRMSE score for all of the station-season combinations. For all data-sets, Abdel et al. [1], Li et al. [26], and Rana et al. [39] produced higher nRMSE compared to DSS-LSTM. Also, DSS-LSTM has achieved lowest mean rank compared to others.

In paper [13], it has been observed that for forecasting solar irradiation 1–6 h ahead, locations with less variability of solar irradiation, ARIMA, and MLP performed better with nRMSE score varying from 18.35% to 33.69% and from 18.26% to 33.84%. On the other side, the locations with high variability of solar irradiation, Bagged Regression Tree, and RF have performed better with nRMSE scores varying from 28.80% to 47.52% and from 28.76% to 48.34%. In paper [11], the authors have reported with overall 9.75% of nRMSE score. In our work, DSS-LSTM has achieved 2.25% of nRMSE. Therefore, the result shows that DSS-LSTM produces better or very competitive results over the papers [11, 13] with a substantially lower value of nRMSE.

6 Conclusion

A stable short-term forecasting model for solar energy generation is critical as there is a lot of variance due to the sub-hourly cloud phenomenon. The proposed LSTM network model is designed to be part of a grid integration software platform that produces 25 and 30 steps ahead reliable forecasts for grid operators and other stakeholders to use in the energy management system. In our current work, we have performed an empirical investigation based on data from three solar stations from two climatic zones of India over two seasons for intra-hour short-term solar forecasting using the LSTM network. Some of our key recommendations for a better LSTM design from our study are as follows:

- *Pre-processing* Using raw data in the case of solar forecasting, LSTM has been able to capture the average variability in predictions by 99% in terms of Explained Variability Score. In comparison, the average variability explained by LSTM applied to pre-processed data is 88%. Thus, we do not need to pre-process data to remove seasonality.
- *Supervised or Non-Supervised* LSTM has performed better when we have preserved the order of the input data. Further to that, stateful LSTMs have produced better performance compared to the stateless LSTM.
- *Batch Size* It has been observed that the nRMSE decreases as we increase the batch size for stations with low variability in GHI, whereas, for the two stations where variability is high, the nRMSE decreases then saturates at a batch size of 36.
- *Effect of prediction horizon* For winter data, 25 and 30 steps ahead prediction leads to nRMSE increase by 8.40% and 13.55% as compared to 20 steps ahead prediction for the DSS-LSTM. For the rainy season, the nRMSE of DSS-LSTM has correspondingly increased by 25.03% and 50.21%.
- *Input data variability and model complexity* It has been observed that input data variability and model complexity are associated. Howrah-Rainy, Ajmer-Rainy need twice the number of nodes compared with the other four station-season combinations because of the variability in GHI.
- *Comparison to existing methods* DSS-LSTM has outperformed Rana et al. [39], Abdel et al. [1], and Li et al. [26] by 52.20%, 15.83%, and 36.09% as measured by nRMSE. This model is also better in terms of mean rank.

For identifying a better design of LSTM networks, this work can be extended by including more input variables, including solar stations from other climatic zones.

Acknowledgements This article results 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).

Compliance with ethical standards

Conflict of interest On behalf of all authors, the corresponding author states that there is no conflict of interest.

References

1. Abdel-Nasser M, Mahmoud K (2019) Accurate photovoltaic power forecasting models using deep lstm-rnn.

Neural Comput Appl 31(7):2727–2740. https://doi.org/10.1007/s00521-017-3225-z

2. Alfadda A, Rahman S, Pipattanasomporn M (2018) Solar irradiance forecast using aerosols measurements: a data driven approach. Solar Energy 170:924–939. https://doi.org/10.1016/j.solener.2018.05.089
3. Alzahrani A, Shamsi P, Dagli C, Ferdowsi M (2017) Solar irradiance forecasting using deep neural networks. Proc Comput Sci 114:304–313. https://doi.org/10.1016/j.procs.2017.09.045
4. Bauer P, Thorpe A, Brunet G (2015) The quiet revolution of numerical weather prediction. Nature 525(7567):47–55. https://doi.org/10.1038/nature14956
5. Benali L, Notton G, Fouilloy A, Voyant C, Dizene R (2019) Solar radiation forecasting using artificial neural network and random forest methods: application to normal beam, horizontal diffuse and global components. Renew Energy 132:871–884. https://doi.org/10.1016/j.renene.2018.08.044
6. Borna K, Ghanbari R (2019) Hierarchical lstm network for text classification. SN Appl Sci 1(9):1124. https://doi.org/10.1007/s42452-019-1165-1
7. Caballero R, Zarzalejo LF, Otero Á, Piñuel L, Wilbert S (2018) Short term cloud nowcasting for a solar power plant based on irradiance historical data. J Comput Sci Technol. https://doi.org/10.24215/16666038.18.e21
8. Caldas M, Alonso-Suárez R (2019) Very short-term solar irradiance forecast using all-sky imaging and real-time irradiance measurements. Renew Energy 143:1643–1658. https://doi.org/10.1016/j.renene.2019.05.069
9. Chiu JP, Nichols E (2016) Named entity recognition with bidirectional lstm-cnns. Trans Assoc Comput Linguist 4:357–370. https://doi.org/10.1162/tacl_a_00104
10. Ding N, Ma H, Gao H, Ma Y, Tan G (2019) Real-time anomaly detection based on long short-term memory and gaussian mixture model. Comput Electr Eng 79:106458. https://doi.org/10.1016/j.compeleceng.2019.106458
11. Feng C, Cui M, Lee M, Zhang J, Hodge BM, Lu S, Hamann HF (2017) Short-term global horizontal irradiance forecasting based on sky imaging and pattern recognition. In: 2017 IEEE Power & Energy Society General Meeting, IEEE, pp 1–5. https://doi.org/10.1109/PESGM.2017.8274480
12. Fliess M, Join C, Voyant C (2018) Prediction bands for solar energy: New short-term time series forecasting techniques. Solar Energy 166:519–528. https://doi.org/10.1016/j.solener.2018.03.049
13. Fouilloy A, Voyant C, Notton G, Motte F, Paoli C, Nivet ML, Guillot E, Duchaud JL (2018) Solar irradiation prediction with machine learning: forecasting models selection method depending on weather variability. Energy 165:620–629. https://doi.org/10.1016/j.energy.2018.09.116
14. Gensler A, Henze J, Sick B, Raabe N (2016) Deep learning for solar power forecastingan approach using autoencoder and lstm neural networks. In: 2016 IEEE international conference on systems, man, and cybernetics (SMC), IEEE, pp 002858–002865. https://doi.org/10.1109/SMC.2016.7844673
15. Gulli A, Pal S (2017) Deep learning with Keras. Packt Publishing Ltd, Birmingham
16. Hochreiter S (1998) The vanishing gradient problem during learning recurrent neural nets and problem solutions. Int J Uncertainty Fuzziness Knowl Based Syst 6(02):107–116. https://doi.org/10.1142/S0218488598000094
17. Hochreiter S, Schmidhuber J (1997) Long short-term memory. Neural Comput 9(8):1735–1780. https://doi.org/10.1162/neco.1997.9.8.1735
18. Huang X, Zhang C, Li Q, Tai Y, Gao B, Shi J (2020) A comparison of hour-ahead solar irradiance forecasting models based on lstm network. Math Probl Eng. https://doi.org/10.1155/2020/4251517
19. Iqbal M (2012) An introduction to solar radiation. Elsevier, Amsterdam
20. Kang J, Jang S, Li S, Jeong YS, Sung Y (2019) Long short-term memory-based malware classification method for information security. Comput Electr Eng 77:366–375. https://doi.org/10.1016/j.compeleceng.2019.06.014
21. Kashyap Y, Bansal A, Sao AK (2015) Solar radiation forecasting with multiple parameters neural networks. Renew Sustain Energy Rev 49:825–835. https://doi.org/10.1016/j.rser.2015.04.077
22. Kumar A, Gomathinayagam S, Giridhar G, Mitra I, Vashistha R, Meyer R, Schwandt M, Chhatbar K (2014) Field experiences with the operation of solar radiation resource assessment stations in india. Energy Procedia 49:2351–2361. https://doi.org/10.1016/j.egypro.2014.03.249
23. Kumari P, Toshniwal D (2020) Extreme gradient boosting and deep neural network based ensemble learning approach to forecast hourly solar irradiance. J Cleaner Prod 279:123285. https://doi.org/10.1016/j.jclepro.2020.123285
24. Laptev N, Yosinski J, Li LE, Smyl S (2017) Time-series extreme event forecasting with neural networks at uber. Int Conf Mach Learn 34:1–5
25. LeCun Y, Bengio Y, Hinton G (2015) Deep learning. Nature 521(7553):436–444. https://doi.org/10.1038/nature14539
26. Li G, Wang H, Zhang S, Xin J, Liu H (2019) Recurrent neural networks based photovoltaic power forecasting approach. Energies 12(13):2538. https://doi.org/10.3390/en12132538
27. Morales-Salinas L, Cárdenas-Jirón LA, González-Rodríguez E (2007) A simple physical model to estimate global solar radiation in the central zone of chile. Department of Environmental Sciences and Natural Renewable Resources, Faculty of Agronomy, University of Chile: Santiago, Chile
28. Mukherjee A, Ain A, Dasgupta P (2018) Solar irradiance prediction from historical trends using deep neural networks. In: 2018 IEEE International Conference on Smart Energy Grid Engineering (SEGE), IEEE, pp 356–361. https://doi.org/10.1109/SEGE.2018.8499394
29. Nikitidou E, Zagouras A, Salamalikis V, Kazantzidis A (2019) Short-term cloudiness forecasting for solar energy purposes in greece, based on satellite-derived information. Meteorol Atmosph Phys 131(2):175–182. https://doi.org/10.1007/s00703-017-0559-0
30. Olah C (2015) Understanding lstm networks. https://colah.github.io/posts/2015-08-Understanding-LSTMs/
31. Perez R, Lorenz E, Pelland S, Beauharnois M, Van Knowe G, Hemker K Jr, Heinemann D, Remund J, Müller SC, Traunmüller W et al (2013) Comparison of numerical weather prediction solar irradiance forecasts in the us, canada and europe. Solar Energy 94:305–326. https://doi.org/10.1016/j.solener.2013.05.005
32. Perveen G, Rizwan M, Goel N (2018) Intelligent model for solar energy forecasting and its implementation for solar photovoltaic applications. J Renew Sustain Energy 10(6):063702. https://doi.org/10.1063/1.5027824
33. Perveen G, Rizwan M, Goel N (2019a) An anfis-based model for solar energy forecasting and its smart grid application. Eng Rep 1(5):e12070. https://doi.org/10.1002/eng2.12070
34. Perveen G, Rizwan M, Goel N (2019b) Comparison of intelligent modelling techniques for forecasting solar energy and its application in solar pv based energy system. Energy Syst Integr 1(1):34–51. https://doi.org/10.1049/iet-esi.2018.0011
35. Perveen G, Rizwan M, Goel N (2019c) Short-term pv power forecasting based on sky-conditions using intelligent modelling techniques. Int J Eng Sci Technol 11(4):49–57

36. Philipp G, Song D, Carbonell JG (2017) The exploding gradient problem demystified-definition, prevalence, impact, origin, tradeoffs, and solutions. arXiv:171205577
37. Prăvălie R, Patriche C, Bandoc G (2019) Spatial assessment of solar energy potential at global scale. A geographical approach. J Cleaner Prod 209:692–721. https://doi.org/10.1016/j.jclepro.2018.10.239
38. Qing X, Niu Y (2018) Hourly day-ahead solar irradiance prediction using weather forecasts by lstm. Energy 148:461–468. https://doi.org/10.1016/j.energy.2018.01.177
39. Rana M, Rahman A (2020) Multiple steps ahead solar photovoltaic power forecasting based on univariate machine learning models and data re-sampling. Sustain Energy Grids Netw 21:100286. https://doi.org/10.1016/j.segan.2019.100286
40. Rathore PKS, Rathore S, Singh RP, Agnihotri S (2018) Solar power utility sector in india: challenges and opportunities. Renew Sustain Energy Rev 81:2703–2713. https://doi.org/10.1016/j.rser.2017.06.077
41. Reikard G (2009) Predicting solar radiation at high resolutions: a comparison of time series forecasts. Solar Energy 83(3):342–349
42. Reikard G, Hansen C (2019) Forecasting solar irradiance at short horizons: frequency and time domain models. Renew Energy 135:1270–1290. https://doi.org/10.1016/j.renene.2018.08.081
43. Ryu A, Ito M, Ishii H, Hayashi Y (2019) Preliminary analysis of short-term solar irradiance forecasting by using total-sky imager and convolutional neural network. In: 2019 IEEE PES GTD Grand International Conference and Exposition Asia (GTD Asia), IEEE, pp 627–631. https://doi.org/10.1109/GTDAsia.2019.8715984
44. Sagheer A, Kotb M (2019) Time series forecasting of petroleum production using deep lstm recurrent networks. Neurocomputing 323:203–213. https://doi.org/10.1016/j.neucom.2018.09.082
45. Soltau H, Liao H, Sak H (2016) Neural speech recognizer: Acoustic-to-word lstm model for large vocabulary speech recognition. arXiv:161009975
46. Sundermeyer M, Schlüter R, Ney H (2012) Lstm neural networks for language modeling, in thirteenth annual conference of the international speech communication association
47. Voyant C, Notton G, Kalogirou S, Nivet ML, Paoli C, Motte F, Fouilloy A (2017) Machine learning methods for solar radiation forecasting: a review. Renew Energy 105:569–582. https://doi.org/10.1016/j.renene.2016.12.095
48. Wang GC, Ratnam E, Haghi HV, Kleissl J (2019) Corrective receding horizon ev charge scheduling using short-term solar forecasting. Renew Energy 130:1146–1158. https://doi.org/10.1016/j.renene.2018.08.056
49. Xingjian S, Chen Z, Wang H, Yeung DY, Wong WK, Woo Wc (2015) Convolutional lstm network: A machine learning approach for precipitation nowcasting. In: Advances in neural information processing systems, pp 802–810
50. Yagli GM, Yang D, Srinivasan D (2019) Automatic hourly solar forecasting using machine learning models. Renew Sustain Energy Rev 105:487–498. https://doi.org/10.1016/j.rser.2019.02.006
51. Yang D, Sharma V, Ye Z, Lim LI, Zhao L, Aryaputera AW (2015) Forecasting of global horizontal irradiance by exponential smoothing, using decompositions. Energy 81:111–119. https://doi.org/10.1016/j.energy.2014.11.082
52. Zhao Z, Chen W, Wu X, Chen PC, Liu J (2017) Lstm network: a deep learning approach for short-term traffic forecast. IET Intell Transp Syst 11(2):68–75. https://doi.org/10.1049/iet-its.2016.0208