Multimedia Tools and Applications
https://doi.org/10.1007/s11042-021-11025-5

1197: ADVANCES IN SOFT COMPUTING TECHNIQUES FOR VISUAL INFORMATION-BASED SYSTEMS

Visualizing solar irradiance data in ArcGIS and forecasting based on a novel deep neural network mechanism

Banalaxmi Brahma[1] · **Rajesh Wadhvani**[1]

Received: 12 October 2020 / Revised: 7 April 2021 / Accepted: 5 May 2021 /
Published online: 07 June 2021

Abstract

Solar power plants are growing tremendously to manage the ever-growing demand for power production and supply in a sustainable manner. Solar irradiance forecasting, being a time series problem, can aid in the planning and design of solar power plants. This work have developed a deep learning forecast model for accurately predicting future values of solar irradiance for given location. The locations have been selected with the help of visual information, which indicates the intensity of solar irradiance. This visual information has been provided by NASA's Prediction of Worldwide Energy Resources (POWER) project. The proposed model utilizes convolutional layers for extraction of internal representations of solar irradiance time-series data along with the attention-based LSTM network for the identification of temporal dependencies. The study is conducted on solar datasets from two locations for a period of 36 years taken from NASA's Prediction of Worldwide Energy Resource (POWER) project archive of Renewable Energy. Experiments were conducted with comparisons against various deep learning models. From the study conducted, we analyze the effect of different factors, such as model, optimizer, horizon, to decide upon the final forecast result. It has been observed that the proposed model performed with an R^2 score of more than 50 percent which indicate excellent forecast performance. Experimental analysis is also conducted for bias and variance as performance evaluators along with other indicators. The bias and variance components also show superior performance of the proposed model. The convolutional and the attention components enhance the performance of the LSTM model with the bias variance mechanism providing improved generalizability.

Keywords Recurrent neural network · Convolutional neural network · Long short term memory · Gated recurrent unit · Attention mechanism · Forecast · Bias · Variance

✉ Banalaxmi Brahma
banalaxmi.173112001@manit.ac.in

[1] Department of Computer Science and Engineering, Maulana Azad National Institute of Technology, Bhopal, Madhya Pradesh, India

1 Introduction

Drastic climate changes and higher electricity demand has lead to the requirement of renewable sources for power generation. Solar energy is a clean, and sustainable energy resource that has become a significant source of energy in recent times [20]. For harnessing this solar power, photovoltaic (PV) or concentrated power plants are used. In this paper, the PV power plants are considered whose performance depends upon electrical parameters, characteristics of the installation as well as meteorological conditions [3]. The major meteorological factor having an impact on the power produced is the solar irradiance. Amrouche et al. [4] states the existence of linear correlation between the maximum PV module power and solar irradiance. The intensity of solar irradiance is dependent on the time and location along with the panel orientation [32]. Solar power, therefore, has intermittent and chaotic behavior. This study aims to forecast solar irradiance in a generalized manner optimally.

The POWER project of NASA was initiated for improving the current Surface Meteorology and Solar Energy (SSE) dataset and creation of new datasets from satellite systems and forecast modeling data. The POWER Data Access Viewer utilizes the ArcGIS World Geocoding Service for providing us with an interface to visualize and access data all around the globe. ArcGIS is a platform which is utilized by organizations for creation, management, sharing and analysis of spatial data and the geocoding services include the conversion of an address to an (x,y) coordinate of latitude and longitude and append results to an existing database record which might also involve mapping. Here, we tend to visualize, explore and forecast the solar related parameters provided in the form of a Data Access Viewer, which was basically formulated for assessing and designing renewable energy systems.

Forecasting techniques are of three types basically, Image-based, Numerical Weather Prediction, and the Statistical & Machine Learning methods that range for predictions of different horizons. The different forecast mechanisms are discussed in Table 1. Solar irradiance data is sequential data ranging for a period of time. However, solar irradiance has variable and complex data characteristics, both spatial and temporal, making its accurate forecasting a challenging task.

As spatio-temporal correlated data is increasing, the need for further research with the consideration of spatio-temporal correlation is much necessary for the task of forecasting solar irradiance [25]. The rational combination of the two state-of-the-art models need to be further exploited for integration of their respective merits [22]. As such, this paper proposes a hybrid model of CNN-LSTM in order to take full advantage of the CNN for spatial feature extraction along with the superiority of the LSTM network for long-term dependencies and time-series features. Here, we explore the capability of the attention mechanism with the advantages of LSTM and CNN to propose a novel solar irradiance forecast mechanism.

Recent studies include work that has implemented deep learning for the task of solar irradiance forecasting. Greff et al. [18] performs an analysis of the different variants of LSTM architecture, concluding that standard vanilla LSTM architecture performs the best amongst all. Srivastava et al. [39] states LSTM as a powerful approach for forecasting using it to predict day-ahead solar irradiance. LSTM model proves to be robust and outperforms other machine learning methods. Qing et al. [35] proposed a solar prediction mechanism by utilizing weather forecast data. The LSTM model was compared to linear least square regression, persistence algorithm, and multilayered feedforward neural networks to predict solar irradiance, with the LSTM model performing with the best accuracy. Wang et al. [42] developed a LASSO-LSTM predicting short term solar intensity with high precision.

Table 1 Overview of Forecasting Techniques

Method	Description	Comment
Autoregressive	Consists of Autoregression(AR) [41], Moving Average Model(MA), ARX [33], ARMA [7], ARMAX [30], ARIMA [46], SARIMA [11], ARIMAX [24], SARIMAX [1] and Generalized Autoregressive Score (GAS) [12, 34] models. Linear over the previous inputs and well understood, easy to compute and provide stable forecasts.	Major limitation is their pre-assumed linearity form of the data that cannot capture complex non-linear patterns.
Neural Networks	Work on non linear transforming layers and are also strongly capable in finding out the complex structures within data. It can reconstruct a data-driven noisy system, which can be used for forecasting.	Data does not require to be necessarily stationary.
Recurrent Neural Networks	Type of neural network architecture accounting for data node dependencies by preserving sequential information in an inner state, which allows the persistence of knowledge accrued from subsequent time steps.	Recurrent Neural Networks can be divided into traditional memory based models and the attention-based ones. Some of the memory based models are LSTM, GRU, Bidirectional RNNs etc.
Long Short Term Memory Networks	Hochreiter et al. [21] proposed the Long Short Term Memory which is an extension of the recurrent neural network architecture by replacement of the conventional perceptron architecture having a memory cell and a gating mechanism regulating information flow across the network.	Most widely used model for the task of solar irradiance forecasting.
Attention	The attention-based models consists of attention LSTM, Self Attention Generative Adversarial Networks and Multi Headed LSTM. The attention vector is based on correlation values and then taking these values' sum weighted by the attention vector.	The attention mechanism which was traditionally introduced and used specifically for machine translation has been recently used for various time series forecasting tasks.
Convolutional Neural Networks	CNN's work successfully to capture various important signal patterns for enhancing the learning ability of the model by applying CNN filters on the observation vectors [38].	The Convolutional Neural Networks (CNN) are also being used for time series forecasting with 1D CNN mainly used for sequence data processing [29].

Various hybrid mechanisms have also been applied to utilize the traditional statistical techniques and the neural network models [19]. The CNN-based forecast mechanisms are also being developed recently for short-term forecasting [13, 44]. The works suggest the

efficiency of LSTM and CNN for short-term forecasts. Recently, the attention mechanism has also been applied for wind speed and solar irradiance forecast modeling [8, 10]. However, none of the mentioned research considered CNN based forecasting technique with the attention-based LSTM model and a bias-variance evaluation strategy for solar irradiance forecasting. Our research contribution focuses on exploiting the convolutional and attention-based LSTM layers for extracting the internal representations and temporal dependencies. Therefore, here we propose the CNN-Attention LSTM architecture for the solar time series forecasting problem.

Furthermore, however strongly capable the neural network architecture be, its efficiency depends on how well it is able to learn and generalize. The present literature only considers the overall performance measure, such as MSE, for validating the forecast performance of models. Often, this overall measure is not enough since it doesn't provide an insight into the inner mechanics of model complexity, misspecification, and data adequacy. In simpler words, model MSE alone cannot be held accountable for the overall learning and generalizability of the model. As such, bias-variance analysis was carried out on traditional neural network architecture and on synthetic datasets to improve upon the generalization capability [6, 40]. MSE decomposition to bias and variance can provide new error metrics for validating model capacity. This paper does an empirical investigation of bias and variance behavior for solar irradiance forecast strategy. Various architectures with different network configurations are analyzed using this mechanism which gives an efficient way for deciding upon a generalized model. The main contributions of this paper include:

1. Visualization and analysis of the POWER repository data for target locations provided by NASA for global solar parameters.
2. Proposal of a hybrid solar irradiance forecast mechanism utilizing CNN, LSTM, and Attention methods.
3. Development of a Bias-Variance performance evaluation method for efficient forecast evaluation.
4. Experimental study for the techniques as mentioned above on datasets from two locations for a period of 36 years from 1983 to 2019 with strength and weakness comparison of model development and evaluation strategy from the perspective of a bias-variance model developed.

The aim is to propose a novel solar forecast mechanism and conduction of a systematic investigation for the reliable forecast from the perspective of model bias and variance. The importance of our proposed method is that it can help to enhance time-series forecasting of solar irradiance data in a more cost-effective manner. Model learning, generalization, and data sensitivity are focused on the framework introduced to reveal the complexity, data adequacy, and misspecification. This provides an additional advantage compared to the existing methodologies that can be implemented in real-world scenarios for solar power plant planning and deployment.The rest of the paper is organized as follows. Section 2 presents the proposed method for model development using the hybrid technology and the model realization using the bias-variance method. The research design and methodology is discussed in Section 3 for an optimal model development followed by the bias-variance analysis. Section 4 presents and analyses the experiments' results, and finally, the paper is concluded in Section 5.

2 Model building

This section presents the building blocks of our proposed model. After giving an overview of the various memory-based and attention-based RNN architectures, the optimizers analyzed in our study and the learning rate tuning mechanism are also discussed.

2.1 Overview of forecast models

Recurrent Neural Network and its advanced variants are described here. The vanilla RNN, LSTM, Bidirectional LSTM, GRU, and attention mechanisms are presented below.

2.1.1 Recurrent neural networks

Rumelhart et al's. [37] work formed the basis of Recurrent Neural Networks. They model the aspect of time by creation of cycles in the network accounting for data nodes' dependencies. The sequential information is preserved in an inner state allowing the knowledge to persist in subsequent time steps.

$$\begin{aligned} Z_t &= W_{XH} X_t + W_{HH} H_{t-1} \\ H_t &= \phi(Z_t) \\ y_t &= W_{HY} H_t \\ Y_t &= \sigma(y_t) \end{aligned} \tag{1}$$

Figure 1 represents an RNN cell with present input denoted as X_t. H_{t-1} is the previous state, W_{XH} is the weight from input to the hidden unit while W_{HH} is the weight between hidden to hidden unit or recurrent weight, and b is the bias. Z_t represents the output before the activation function ϕ is applied. Then, H_t is the output of the hidden unit used in the

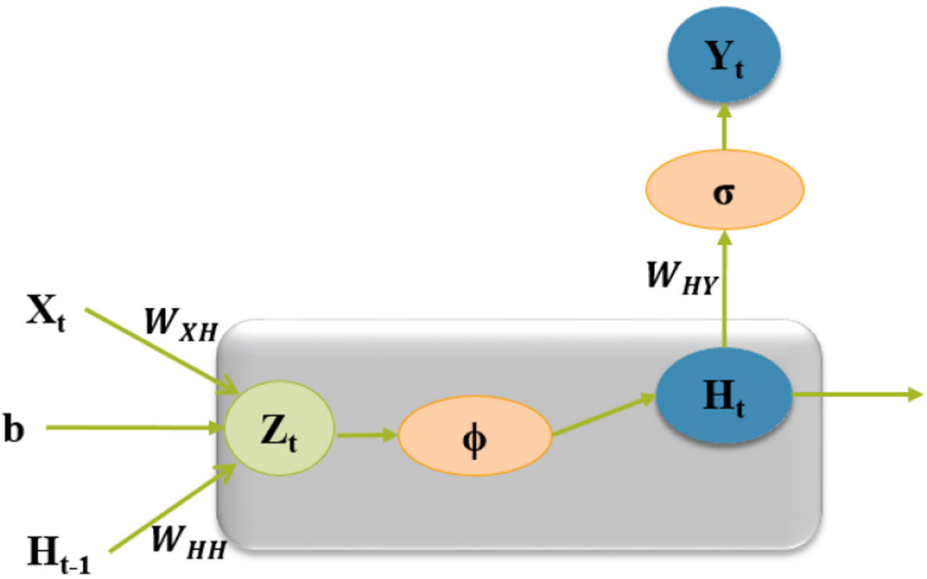

Fig. 1 RNN cell

final output computation of that RNN unit. Computation of the final output Y_t is done by application of another activation function to the output of the hidden unit, with W_{HY} being the weight between hidden to the output unit. Application and selection of the activation function are dependent on the task being performed [16].

2.1.2 Long short term memory

Hochreiter et al. [21] proposed the LSTM architecture that improves upon the efficiency of traditional sequence learning mechanisms. The vanishing and exploding gradients problem lead to the development of LSTM. LSTM, as represented in Fig. 2, introduce additional gating components to RNN, the input, forget and output gate. The forward pass equation is stated in (2).

$$\begin{aligned} A_t &= tanh(W_{cur} X_t + R_{cur} H_{t-1}) \\ I_t &= sigmoid(W_{inp} X_t + R_{inp} H_{t-1}) \\ F_t &= sigmoid(W_{for} X_t + R_{for} H_{t-1}) \\ O_t &= sigmoid(W_{out} X_t + R_{out} H_{t-1}) \\ C_t &= I_t \odot A_t + F_t \odot C_{t-1} \\ H_t &= O_t \odot tanh(C_t) \end{aligned} \tag{2}$$

A_t processed the previous state and current input after which I_t, the input gate, makes the decision on which parts of A_t are to be added to C_t, the long term state. tanh and sigmoid are the activation functions. The forget gate F_t has the responsibility to decide on parts of C_{t-1} to be erased in order to discard unnecessary parts. O_t is the output gate that finds the parts to be shown as output. Therefore, H_t is the short-term state being shared between cells, and C_t is the long-term state in which memories are dropped as well as added.

2.1.3 Bidirectional long short term memory

LSTM processes inputs in a strict order, temporally. The current input has previous inputs' context but not the future. Graves et al. [17] introduced the Bidirectional LSTM model for addressing this. It duplicates the processing chain of LSTM such that input processing can

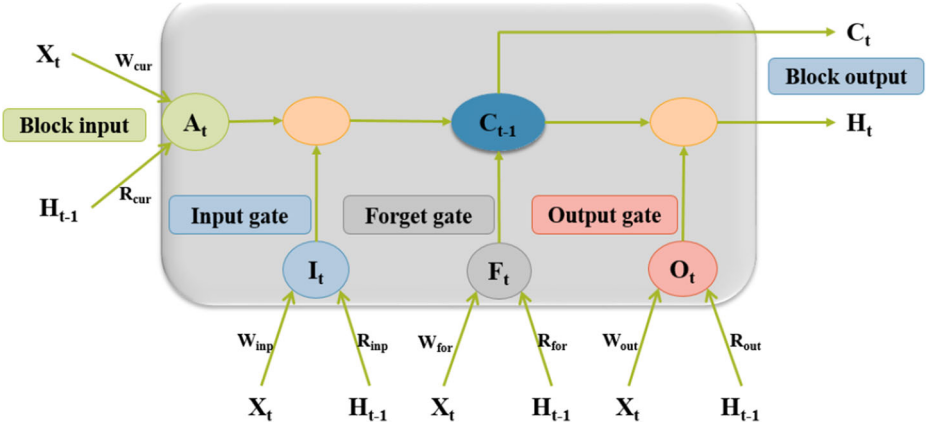

Fig. 2 LSTM cell

be performed in forward as well as reverse time order. The network is now capable of getting an insight into the future context.

2.1.4 Gated recurrent unit

Kyunghyun et al. [27] proposed the Gated Recurrent Unit as a simplified LSTM version in which explicit use of cell states is not performed. Both state vectors are merged into a single vector for simplification. Figure 3 presents a GRU cell, and the (3) denotes the forward pass.

$$\begin{aligned} z_t &= sigmoid(W_z x_t + R_z h_{t-1}) \\ r_t &= sigmoid(W_r x_t + R_r h_{t-1}) \\ g_t &= tanh(W_g x_t + R_g (r_t \otimes h_{t-1}) \\ h_t &= (1 - z_t) \otimes tanh(W h_{t-1} + z_t \otimes g_t) \end{aligned} \tag{3}$$

It is an LSTM variant used frequently. In Fig. 3, r_t denotes the reset gate, z_t the update gate and g_t the candidate hidden state. No output gate is present and full state vector is output in every time step. Finally, we get the output hidden state h_t given the input h_{t-1}. However, a new gate controller is introduced which controls part of the previous state to be shown to the main layer.

2.1.5 Attention mechanism

Bahdanau et al. [5] presented the attention-based mechanism belonging to the sequence to sequence model, built for the task of neural machine translation. An encoder and decoder is present in which the encoder encodes the input to a vector of fixed length and the decoder translates it [23]. However, the performance of this sequence to sequence model deteriorates with an increase in input sentence length. Therefore, for the purpose of remembering long

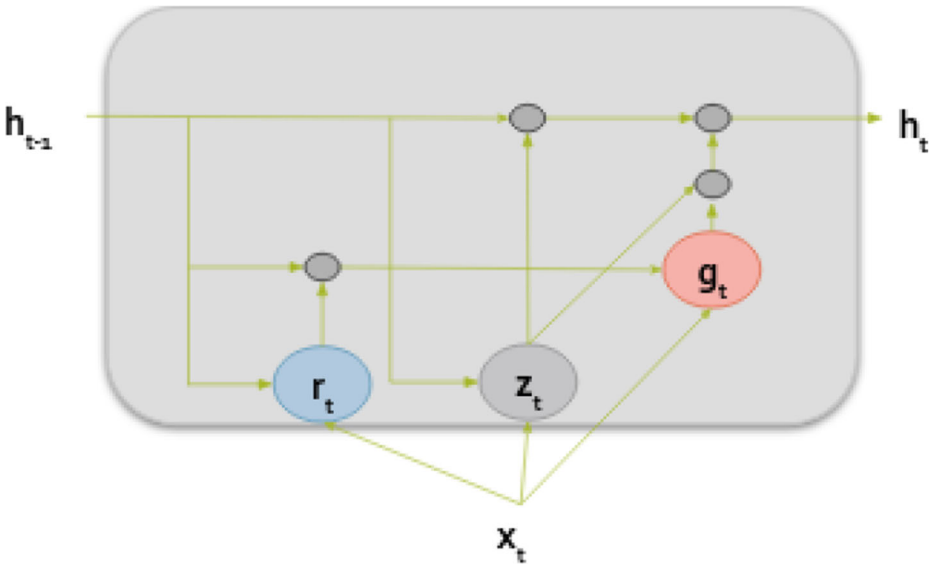

Fig. 3 GRU cell

inputs, mainly sentences for translation, the extension to align and translate jointly was proposed.

The decoder receives the hidden state of the last encoder, which is a vector representation. Even for a long input, the decoder would receive this one vector for prediction of output, which in turn leads to forgetting. This leads to the introduction of attention which acts as an interface between the encoder and decoder. It provides information from every encoder's hidden state to the decoder. The model can now focus on useful parts of the input sequence selectively based on the scoring function [9].

Score denotes a scoring function that denotes relevance between the input vectors. The context vector is then computed, which is forwarded to the decoder layer [38].

$$score(h_{t-1}, s_{t-1}) = v_a^T \tanh(W_a[h_{t-1}; s_t] + W_x x + b_a) \tag{4}$$

$$\alpha_{t-1} = \frac{\exp(score(h_{t-1}, s_{t-1}))}{\sum_{i=1}^{t} \exp(score(h_{i-1}, s_{i-1}))} \tag{5}$$

$$ContextVector = \sum_{i=1}^{t} \alpha_i h_i \tag{6}$$

Here, x is the given input with h_{t-1} being the hidden state and s_{t-1} the cell state. W_a, $W_x$and b_a are the attention weights and bias respectively. The additional benefit of attention is the relevant information computation after its extraction through scoring function which aides in the management of long-term dependencies.

2.2 Overview of optimization techniques

The main goal of optimization is minimizing a loss function. This section enlists and describes the various optimization techniques widely used in LSTM network training along with their weight update mechanism.

2.2.1 AdaGrad

Duchi et al. [15] proposed the AdaGrad optimizer that works on the learning rate component. It utilizes gradient squares aggregation and selects the learning rate for each parameter according to gradient size. Here, S is the cumulative sum of current and past squared gradients.

$$w_{t+1} = w_t - \frac{\alpha}{\sqrt{S_t + \epsilon}} \cdot \frac{\partial L}{\partial w_t} \tag{7}$$

$$S_t = S_{t-1} + [\frac{\partial L}{\partial w_t}]^2 \tag{8}$$

2.2.2 Adadelta

Zeiler et al. [45] introduced the Adadelta, which focuses on the learning rate component. Instead of accumulating past gradients, it uses fixed-sized windows. Adadelta deletes the learning rate parameter by its replacement with D, the exponential moving average of squared deltas.

$$w_{t+1} = w_t - \frac{\sqrt{D_{t-1} + \epsilon}}{\sqrt{S_t + \epsilon}} \cdot \frac{\partial L}{\partial w_t} \tag{9}$$

$$D_t = \beta D_{t-1} + (1-\beta)[\Delta w_t]^2 \tag{10}$$

$$S_t = \beta S_{t-1} + (1-\beta)[\frac{\partial L}{\partial w_t}]^2 \tag{11}$$

$$\Delta w_t = w_t - w_{t-1} \tag{12}$$

2.2.3 Adaptive moment estimation

Kingma and Ba [26] proposed the adaptive moment estimation, more widely known as ADAM, which is generally robust to the hyperparameter choice. It consists of the moment vectors m and v being first and second-order of the exponentially weighted gradients, along with its bias corrections, $\hat{m}$, $\hat{v}$.

$$m_t = \beta_1 m_{t-1} + (1-\beta_1)\frac{\partial J}{\partial w_{t-1}} \tag{13}$$

$$\upsilon_t = \beta_2 \upsilon_{t-1} + (1-\beta_2)\frac{\partial J^2}{\partial w_{t-1}} \tag{14}$$

Bias corrections are given:

$$\hat{m_t} = \frac{m_t}{1-\beta_1^t} \tag{15}$$

$$\hat{\upsilon_t} = \frac{\upsilon_t}{1-\beta_2^t} \tag{16}$$

Finally, parameter updation is done as:

$$w_{t+1} = w_t - \alpha \frac{\hat{m_t}}{\sqrt{\hat{v_t}} + \epsilon} \tag{17}$$

ADAM does the computation of adaptive learning rates for every parameter. It stores the exponentially decaying average of past squared gradients along with the exponentially decaying average of past gradients.

2.2.4 AdaMax

ADAM performs update by scaling the gradient inversely proportional to the l_2 norm of past gradients. This can be generalized to the l_p norm. Norms with larger p values exhibit unstable behavior. However, l_∞ shows stability. As such, Kingma and Ba [26] also proposed infinity norm constrained υ_t represented as follows:

$$u_t = \beta_\infty \upsilon_{t-1} + (1-\beta_2)\frac{\partial J^\infty}{\partial w_{t-1}} \tag{18}$$

As such, the parameter update rule can be updated as:

$$w_{t+1} = w_t - \alpha \frac{\hat{m_t}}{u_t} \tag{19}$$

2.2.5 Nadam

Dozat [14] proposes that rather than applying the momentum step twice, now the look-ahead momentum vector can be applied directly to update the current parameters. The Nadam incorporates Nesterov momentum with Adam. Rather than utilizing the previous momentum vector, the current momentum vector is used to look ahead. The previous momentum vector

can be replaced with the current momentum vector for the addition of Nesterov momentum to Adam. The Nadam update rule can be stated as:

$$w_{t+1} = w_t - \frac{\alpha}{\sqrt{\hat{v}_t} + \epsilon}(\beta_1 \hat{m}_t + \frac{(1 - \beta_1 g_t)}{1 - \beta_1^t}) \tag{20}$$

2.2.6 AMSGrad

The adaptive learning methods are at times outperformed by Stochastic Gradient Descent(SGD) methods and do not converge to a solution which is optimal. Exponential averaging diminishes the effects of the informative mini batches which occur rarely leading to poor convergence. As such, AMSGRAD [36] was proposed which does not use the bias corrected υ_t but uses:

$$\hat{\upsilon}_t = max(\hat{\upsilon_{t-1}}, \upsilon_t) \tag{21}$$

The rest of the update rule is similar to that of ADAM, as stated above.

2.3 Learning rate

The learning rate needs to be tuned experimentally since its optimal value is dependent on the architecture of the model and the dataset in use. Yu et al. [43] states that fixed learning rates are less efficient since, an efficient learning algorithm should be able to dynamically vary its learning rate according to the changes of the gradient values. As in [31], a simple warm restart technique to train the deep neural network is given in (22) and have been applied in the proposed model of forecasting. The method periodically simulates warm restarts, wherein each restart, the learning rate is initialized to some value and is scheduled to decrease. It requires fewer epochs than other methods, but more experimentation is required on a variety of network structures.

$$\eta_t = \eta^i_{min} + \frac{1}{2}(\eta^i_{max} - \eta^i_{min})(1 + cos(\frac{T_{cur}}{T_i \pi})) \tag{22}$$

η_t is learning rate at t time step, with η^i_{min} and η^i_{max} as the range of learning rate. $T_{current}$ is the number of epochs, and T_i the number of epochs in a cycle. This mechanism is based on the cosine function, and it can essentially exit local minima continuing to explore the loss landscape.

3 Proposed method

This section presents the proposed hybrid mechanism developed for solar irradiance forecasting. Firstly, the proposed solar irradiance forecast model is discussed. The hybrid CNN-LSTM and the attention-based forecast mechanisms are then described, which constitute the proposed CNN-Attention LSTM model developed. Then, the proposed CNN-Attention LSTM method is presented, the efficiency of which is tested in the later sections. Furthermore, the model realization method using the bias and variance method is also described. The algorithm for calculating bias and variance and the flow of events for using bias and variance as a performance evaluation metric is shown here.

The proposed method in this work consists of a deep learning forecast network and a bias-variance calculation mechanism. Our model differs from the existing mechanisms since, in addition to the LSTM and convolution components, an attention layer is also introduced for the task of solar irradiance forecasting. Conventionally, either the RNN variants or the RNN variants with the convolution layers were only utilized for the task. Also, the performance evaluation criteria only consisted of overall metrics such as MSE, MAE, or RMSE. Here, an additional framework is introduced, which decomposes the MSE into its bias and variance components. This helps in enhancing the model learning and generalizability by providing insights into model misspecification and complexity.

3.1 The proposed model

The ArcGIS data from the solar data repository is firstly analyzed visually utilizing the POWER layers. This provides us with 22 years of climatology average for the solar parameters. After the analysis of these images, the target locations' historical dataset is downloaded. Figure 4 represents the solar irradiance forecasting model structure that has been adopted in the present work. Solar irradiance measurements refers to the 36 years of data which is collected from NASA's POWER project archive. This data is first validated for its missing values and pre-processed to create a training set for the proposed LSTM model. In the training stage, the models are developed and further optimized for accurate forecasting. Previous values of data are input to predict the next day's output. After developing the model, the actual and the predicted next-day values are compared to attain the performance metrics. The various deep learning forecast models are also evaluated utilizing this model structure for validating the performance of the proposed model.

3.1.1 CNN-LSTM

Convolutional Neural Network [28] architecture has three kinds of layers, the convolutional layer, the fully connected layer and the pooling layer. Convolutional layers take in inputs in the form of feature maps from the previous layer and perform a convolution operation

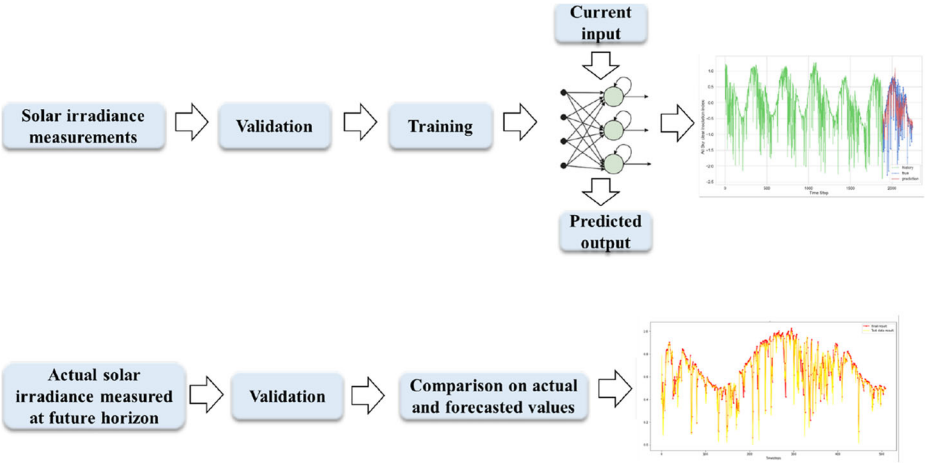

Fig. 4 Solar Irradiance Forecasting Model Structure

between inputs and filters. In time series forecasting, the convolutional layers can extract useful knowledge and learn the internal representations.

$$conv(x, y) = \sum_i w_i v_i \tag{23}$$

In (23), w_i are convolution kernel parameters, v_i is the previous layer output and (x, y) is the spatial coordinate. The feature map is computed with b being scalar bias and g the non-linear activation function denoted in (24).

$$z(x, y) = g(conv(x, y) + b) \tag{24}$$

The method of hybrid CNN-LSTM is constituted by convolutional and LSTM layers. The convolution operation reduces number of parameters while pooling layer is used for the combination of the output of neurons cluster to a single neuron. The pooling layer also decreases computation cost and the number of parameters of the network. LSTM layers are present after the CNN layers in the network.

3.1.2 Attention LSTM

The attention mechanism mainly solves the long-term dependency issue in which historical values from far in the past might affect the present-day forecast. The identification of dynamic inter-dependencies and relevant features can also be performed through attention. The different attention-based mechanisms vary in the encoder-decoder architecture and score function. This work utilizes content-based attention. The attention vectors in cosine scoring [2] or content-based attention are calculated based on the similarity between the key and memory rows. It computes the cosine similarity as denoted in (25), with s_t as the cell state and h_i as the hidden state, which is then normalized by the softmax function.

$$score(s_t, hi) = cosine[s_t, h_i] \tag{25}$$

The encoder layer is constituted by the LSTM, whose outputs are given to an attention layer for computation of score function and later, the context vector. Then, it is forwarded to the Dense layer or fully connected layer, which gives the final output of future solar irradiance data. Attention mechanism is different from the conventional LSTM mechanism since it introduces an additional attention component to the memory based LSTM gating mechanism. It acts as an interface between layers and provide information from every hidden state. The model is capable of selectively focusing on input sequence's useful parts on the basis of scoring function.

3.1.3 Proposed method

The proposed method for solar irradiance forecasting is described here as shown in Fig. 5. This method consolidates the benefits of both the attention and hybrid CNN-LSTM model, as stated above. The three components comprising the proposed model are the CNN, LSTM, and the Attention mechanism. The CNN develops the features from the input time series data by performing complex mathematical operations. Simultaneously, the LSTM layer with the attention mechanism exploits these features to identify dependencies in data. Overall, these three components aid in the robust forecasting of solar irradiance data.

The padding layer after the input, as shown in Fig. 5, performs the zero-padding to control dimension shrinkage due to the application of filters and avoid loss of information. Then, we have the convolution block consisting of the convolution and pooling operations.

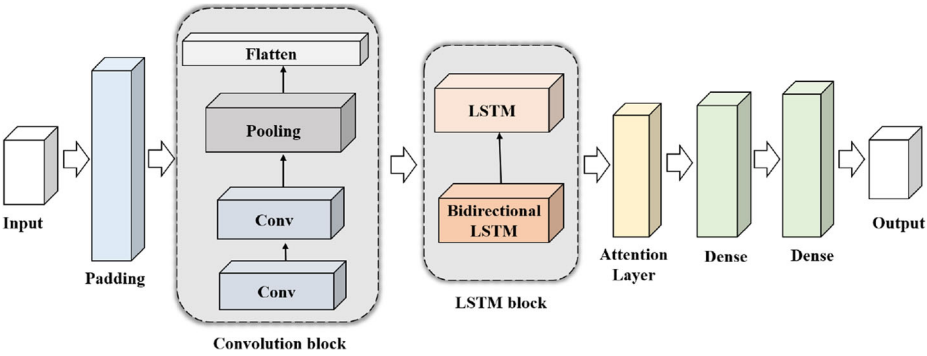

Fig. 5 Proposed Method Architecture

This block is used for the identification of the time series patterns, which is then fed to the LSTM block for temporal processing. The pooling layer reduces variance, computational complexity and also extracts the low-level features. Average Pooling is used here since it encourages the network for identification up to the full extent of the object since loss for this pooling benefits when the network identifies all the discriminative regions of an object, time-series data of solar irradiance in our case.

After the convolution block, the LSTM block works upon the temporal contexts of the pattern features detected by the previous block. The past and future contexts are observed along with memory units and cell states being consolidated for temporal dependencies. The LSTM layers also address the vanishing and exploding gradients. The regularization applied here is dropout regularization. The output from this block is then processed by the attention layer, which uses the content-based scoring function.

The input $(x_1, ..., x_p)$ firstly passes through the convolution block, giving $(y_1, ..., y_p)$ as shown in (26).

$$y_t = Conv(x_t) \tag{26}$$

The LSTM block then processes this output. The generalized form is shown in (2) with h_t as the hidden state output. W_* represents the weights associated with the respective values.

Then, this hidden state is passed to the attention layer as denoted in (27), which uses the current hidden state and the recurrent unit's cell state to calculate the score function S. σ here represents the softmax function.

$$S(h_t, s_t) = \sigma(score(h_t, s_t)) = \sigma(cosine[h_t, s_t]) \tag{27}$$

The context vector C is computed, which is then passed to the next layer. This layer processes the weighted context vector shown in (28). The fully connected output layer finally gives the target variable.

$$C = \sum_{i=1}^{t} S_i h_i \tag{28}$$

The use of the optimization technique does minimization of the loss function. In this work, six different types of optimizers are utilized for the task of solar irradiance forecasting. The various methods are then compared based on the results obtained along with analysis for achieving better forecast accuracy. The optimization techniques include AdaGrad, Adadelta, Adam. AdaMax. Nadam and AMSGrad.

After error propagation and proper model tuning for parameters and hyperparameters, the model is evaluated based on standard performance metrics and the bias-variance mechanism described in the next section.

3.2 Proposed method for model realisation

This study also includes an empirical comparison between different forecast model specifications in terms of bias and variance. Performance improvement of neural network models can be done by the right choice of model and optimizers according to the present data. However, the number of these factors is significant, and it is not always possible to select the optimal one by analyzing the existing performance metrics only. The deep neural network architectures are also prone to the problem of overfitting. To tackle this, here we introduce a bias-variance model. The proposed methodology of designing a bias-variance model for forecasting consists of a model search using bias and variance.

Modeling leads to different forecast variants of the neural network. $F_i(\mathbf{X_n}, \boldsymbol{\tau_m})$ denotes i model variants for test set $\mathbf{X_n}$ given m realizations of training data τ. The losses, MSE, RMSE, R^2 score are computed for all the model variants. Six different optimizers are used for the analysis of performance. ADAM and its variants are mainly chosen for analysis since ADAM is known to have the power of faster convergence and is the most widely used optimizer. The MSE loss function and RMSE and R^2 score are used to analyze the performance. These are used for model training and comparison based on complexity and bias and variance computed. Dropout regularization is used to control overfitting. Also, forecast is performed for different horizons. Their effects are analyzed along with their contribution to overall generalization.

After the model selection phase, a search and validation mechanism is proposed, deciding upon the best model configuration as shown in Fig. 6. Now, calculation of bias and variance also requires several data instances from the same data generating process (DGP). However, bias and variance give the error values for different training instances on the same test set, i.e., it tests for the models' sensitivity to data and its prediction capability. This method is

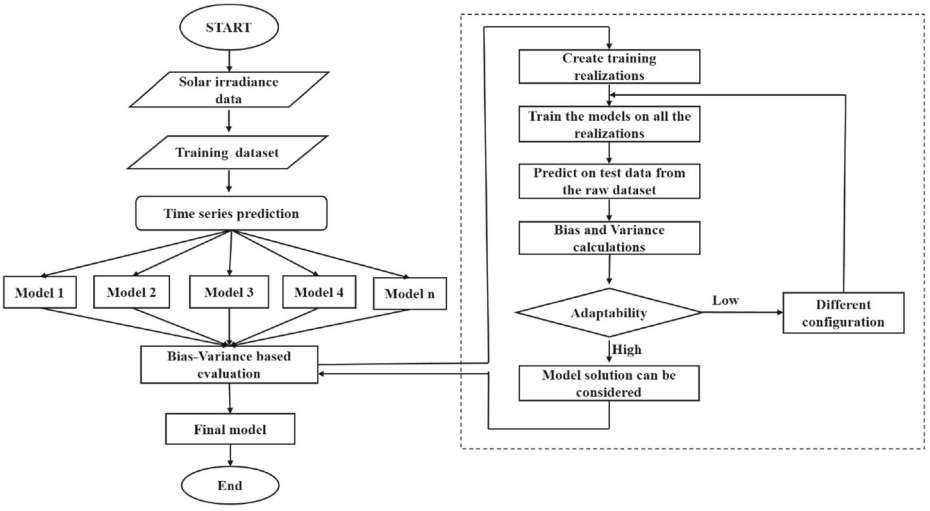

Fig. 6 Bias Variance Calculation Flowchart

similar to the proposed methodology of bias-variance calculation by Berardi and Zhang [6]. However, we tend to implement the calculation framework on real-world datasets instead of a synthetically generated one. Different training instances denote the data in real-world being influenced by noise or the data at hand being not representative of the actual data generating process. In this manner, the model's true potential to forecast can be found out. The algorithm followed for the calculation of bias and variance is stated below.

Algorithm 1 Calculation of Bias and Variance.

Result: Bias, Variance and Error of given model
dats instances=N;
data realizations=M;
target value= $E(D_n \mid \mathbf{X_n})$;
while $M \neq 0$ **do**
    Train the model for particular parameter configuration;
    Find the predicted value=$F(\mathbf{X_n}, \tau_m)$
end
Bias=$\frac{1}{N}\sum_{n=1}^{N}(E(D_n \mid \mathbf{X_n}) - \mathbf{F}(\mathbf{X_n}, \tau_m))^2$;
Variance=$\frac{1}{MN}\sum_{m=1}^{M}\sum_{n=1}^{N}(\vec{F}(\vec{X_n}, \tau_m) - F(\vec{X_n}, \tau_m))^2$;
Error=$\frac{1}{MN}\sum_{m=1}^{M}\sum_{n=1}^{N}(E(D_n \mid \mathbf{X_n}) - F(\mathbf{X_n}, \tau_m))^2$;

In Algorithm 1, $\mathbf{F}(\mathbf{X_n}, \tau_m) = \frac{1}{M}\sum_{m=1}^{M} F(\mathbf{X_n}, \tau_m)$ is the average prediction on input $\mathbf{X_n}$. The test set is the same in all the cases of bias-variance computation, whereas the training instances vary. This variation is created from the data at hand using Gaussian noise. Gaussian noise is used as a result of the Central Limit Theorem. So, we attain different training instances from the same dataset, reflecting the real wold tendency to manipulate with the actual DGP. As such, in Algorithm 1, M denotes the different realizations of the given training dataset. After training the model on these realizations, the actual test set is used for forecasting future values for different horizons. After following this error calculation framework and observing the bias-variance tradeoff, the best performing model is decided upon. The bias term of a model represents the model's capability in discovering the underlying patterns in data, and variance measures the sensitivity of the function to the particular choice of data set. Thus, bias measure reflects learning ability and variance, the generalizability. Hence, this is an addition to the existing forecast methodology enhancing generalization.

In the MSE method, the error calculation is dependent on the particular dataset. Change of dataset may cause a difference in the estimation function and hence estimation error. However, in the bias-variance method, this overall prediction is further decomposed into bias and variance components. These metrics are calculated on different realizations of the same dataset, tested upon a particular test set. In the MSE calculation case, evaluating model fitting and forecasting performance consists of only an overall measure. The MSE reflects the general modeling and forecasting performance, but they do not provide full insight on how and why overfitting arises. It also does not provide insights concerning how to improve forecast performance. However, the decomposed components of bias and variance provide essential statistical properties of the empirical model. Bias measures the error of the forecasting model in learning the underlying patterns in a time series. It reflects the extent to which the estimation function's average function over all possible datasets from the same data generating process differs from the desired function. Variance relates to the stability of models built on different data samples from the same generating process and offers insights into the model's generalizability. It measures the sensitivity of the estimation function to

the training dataset. Therefore, the proposed bias and variance method reveals the model complexity, model misspecification, and data adequacy.

4 Research design and methodology

This section presents the experimental setup for the research performed. The datasets used for simulation and performance estimation model is firstly discussed. After that, the models used for comparison are summarised, and then the performance evaluation metrics are stated.

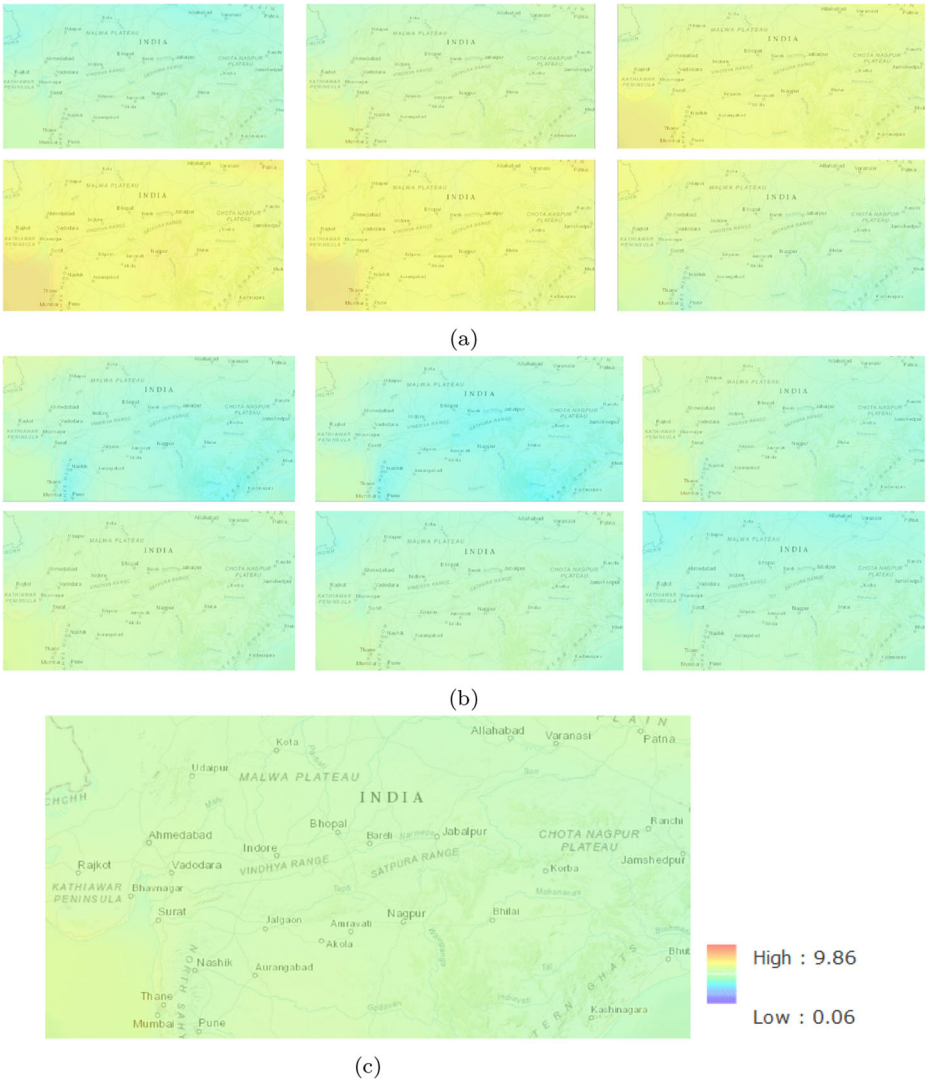

Fig. 7 Visualization of the Solar Irradiance Parameter from the POWER data repository (**a**) 22 Year Climatology Average for the month of January to June (**b**) 22 Year Climatology Average for the month of July to December (**c**) 22 Year Climatology Average annually

4.1 Dataset and preprocessing

The dataset is taken from NASA's POWER project database, which consists of the solar irradiance data, which is represented as all-sky insolation on a horizontal surface in the database having unit $kwhr/m^2$ per day. Figure 7 represents the solar irradiance parameter's layer on the gridded map for a 22-year climatology average. The images first denote the target region's layer for all the months and then the annual average. The range of the solar parameter values is also shown in Fig. 7. This is one of the visual interpretations that can be performed on the ArcGIS World Geocoding Service provided data. As can be observed, the locations selected fall under the same category of the layer. The range of solar irradiance is found to be the same for the 22 years average on both sites for approximately all the months. Since the region selected has three distinct seasons, winter from December to February, summer from March to May, and rainy from June to October, this can be clearly observed

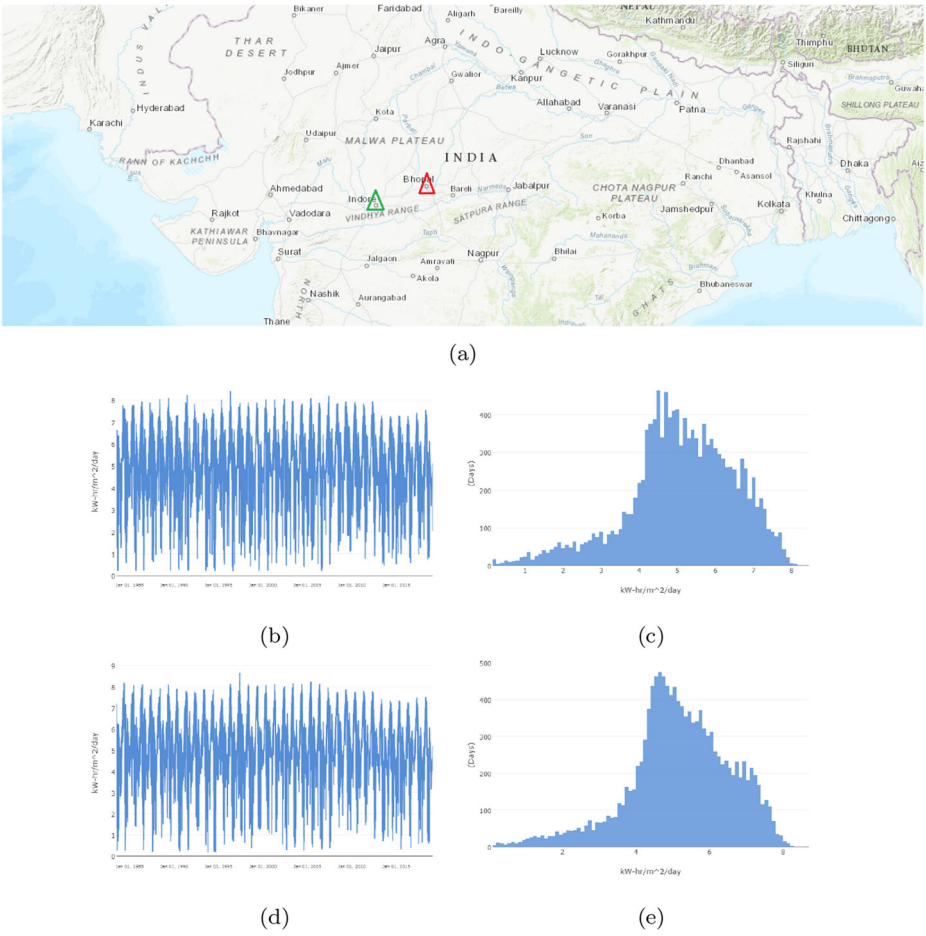

Fig. 8 Data collected from POWER data repository for the two locations on which the case study is performed: (**a**) The two locations with red triangle representing Location 1 and the green triangle representing Location 2, (**b**) Solar Irradiance Data for Location 1, (**c**) Irradiance distribution according to days for Location 1 (**d**) Solar Irradiance Data for Location 2 and (**e**) Irradiance distribution according to days for Location 2

in the variations in the months of different seasons and consistency of same season months of the solar irradiance parameter. Also, the map analysis representing annual climatology denoted the sites as having similar characteristics in terms of overall solar irradiance data. This visual information is helpful in terms of the selection of potential locations for PV solar power plants. As in our case, after this analysis, the two sites are finalized, after which the forecast operation is performed.

Data is collected for two regions: Location 1 with latitude and longitudes 23.25991, 77.41261 (Bhopal, India) and Location 2 with 22.71961, 75.85771 (Indore, India) latitude and longitude for a period of 36 years from 1983 to 2019. Figure 8 represents the locations on which the solar irradiance forecasting is to be performed. The map indicates the repository locations with the red triangle denoting Location 1 and the green triangle pointing Location 2. Table 2 represent the datasets' descriptive statistics to describe the nature of their distribution. The daily solar irradiance data are also shown in Fig. 8 for both locations to illustrate the data pattern and distribution over the days.

The dataset is split into training, validation, and test set according to this performance estimation strategy. Data normalization is performed before model training to standardize the inputs and approach faster towards the global minima. It also ensures that larger inputs do not overwhelm or become dominant.

4.2 Forecast Models

The models used for forecasting and comparison with the proposed model are discussed here. Here, three categories of models have been developed. In the first category, memory-based RNN, i.e., RNN, LSTM, Bidirectional LSTM, and GRU, are discussed. In this category, 100 epochs are used for training the network. The train-test split ratio kept is 80:20. The conventional RNN forward pass is denoted in (1). As an advancement to the RNN, we have the LSTM whose forward pass is shown in (2). Similarly, for GRU, we have used the (3). The hyperparameter learning rate is tuned using the mechanism of warm restarts. The random search was used for the different parameter settings of hidden layers, hidden neurons, and batch size.

In the second category, the CNN-based model for time series forecasting is discussed. The forward pass equation for a CNN-LSTM network is presented in (23). The number of epochs used for training the model is 100 while the train-test split is 80:20. The hyperparameters are tuned in a similar fashion to the memory-based RNN.

In the third category, the attention-based model is discussed. Equations (4-6) represent the forward pass of an attention layer. The learning rate hyperparameter is tuned through the warm restarts mechanism. The number of epochs is 100, and the train-test split ratio is 80:20.

Table 2 Descriptive statistics of dataset

Statistic	Location 1	Location 2
Total observations	13000	13000
Date Range	1983-2019	1983-2019
Standard Deviation	1.42	1.38
Maximum	8.41	8.66
Minimum	0.2	0.16
Mean	5.09	5.17

Table 3 Forecast Modeling Scenarios

Parameter	Variants
Model	RNN, LSTM, GRU, Bidirectional LSTM, CNN-LSTM, Attention-LSTM, Proposed
Optimizers	AdaGrad, Adadelta, Adam. AdaMax, Nadam, AMSGrad
Horizon	One day ahead, Four days ahead, Ten days ahead
Regularization	Dropout
Hyperparameter	Learning Rate with Warm Restarts

All these models have been optimized by the AdaGrad, Adadelta, Adam, AdaMax, Nadam, and AMSGrad optimizers. The related equations are presented in (7-21) respectively. Table 3 below states the comparison scenarios in brief.

Hyperparameters and topologies were optimized for the task of forecasting. The forecast horizon considered here is the number of future days ahead values of solar irradiance the model predicts. The task performed here is that of short-term solar irradiance forecasting. Therefore, previous daily data values are used for forecasting one day, four days ahead, and ten days ahead values.

4.3 Performance metrics

For verification of forecast model performance, the goodness of fit has to be measured. The standard performance metrics consists of MAE, MSE, RMSE, R^2 score, SSE, etc. The metrics shown in Table 4 are used here..

Bias and Variance are also calculated as described in the previous section for different training instances from the same real-world dataset with a common test set. In contrast, the other error metrics are calculated in conventional way. Table 5 represents these metrics for the expected value and the predicted values.

5 Results and discussion

The result analysis of the proposed work for different solar irradiance forecast models is being stated below. The performance metrics obtained after experiments performed on the two datasets are shown. Evaluation of the forecast models for different optimizers and horizons is performed.

Table 4 Performance Evaluation Metrics

Metric	Equation
MSE	$\frac{1}{N}\sum_{n=1}^{N}(E(D_n \mid \mathbf{X_n}) - F(\mathbf{X_n}, \mathbf{w}))^2$
RMSE	$\sqrt{\frac{1}{N}\sum_{n=1}^{N}(E(D_n \mid \mathbf{X_n}) - F(\mathbf{X_n}, \mathbf{w}))^2}$
R^2	$1 - \frac{\sum_{n=1}^{N}(E(D_n \mid \mathbf{X_n}) - F(\mathbf{X_n}, \mathbf{w}))^2}{\sum_{n=1}^{N}(E(D_n \mid \mathbf{X_n}) - \mathbf{F(X_n}, \mathbf{w}))^2}$

Table 5 Bias-Variance Metrics

Metric	Equation
Bias	$\frac{1}{N}\sum_{n=1}^{N}(E(D_n \mid \mathbf{X}_n) - \mathbf{F}(\mathbf{X_n}, \tau_m))^2$
Variance	$\frac{1}{MN}\sum_{m=1}^{M}\sum_{n=1}^{N}(\mathbf{F}(\mathbf{X_n}, \tau_m) - F(\mathbf{X_n}, \tau_m))^2$

Table 6 Performance Evaluation for 1 Horizon on Location 1

Model	MSE	RMSE	R^2
RNN	9.765	9.8818	68.4786
LSTM	9.656	9.8267	68.8298
GRU	9.653	9.8253	68.8381
Bidir-LSTM	9.560	9.7780	69.1378
CNN-LSTM	9.558	9.7769	69.1446
Attention LSTM	9.558	9.7765	69.1469
Proposed	**9.531**	**9.7627**	**69.2339**

Table 7 Performance Evaluation for 4 Horizon on Location 1

Model	MSE	RMSE	R^2
RNN	13.594	11.6597	56.1173
LSTM	13.192	11.4859	57.4153
GRU	13.254	11.5129	57.2151
Bidir-LSTM	13.209	11.4930	57.3627
CNN-LSTM	13.225	11.5002	57.3091
Attention LSTM	13.195	11.4869	57.4083
Proposed	13.111	**11.4504**	**57.6784**

Table 8 Performance Evaluation for 10 Horizon on Location 1

Model	MSE	RMSE	R^2
RNN	15.572	12.4789	49.9787
LSTM	15.490	12.4459	50.2430
GRU	15.297	12.3683	50.8619
Bidir-LSTM	15.165	12.3147	51.2862
CNN-LSTM	15.184	12.3223	51.2269
Attention LSTM	15.238	12.3446	51.0507
Proposed	**15.048**	**12.2673**	**51.6612**

Table 9 Performance Evaluation for 1 Horizon on Location 2

Model	MSE	RMSE	R^2
RNN	8.100	9.0005	71.6343
LSTM	7.911	8.8944	72.2989
GRU	7.855	8.8632	72.4927
Bidir-LSTM	7.794	8.8287	72.7065
CNN-LSTM	7.894	8.8852	72.3560
Attention LSTM	7.795	8.8292	72.7037
Proposed	**7.747**	**8.8020**	**72.8714**

5.1 Comparison of performance metrics

Experiments were performed on datasets from the two locations for all the model variants. The Tables 6, 7, 8, 9, 10 and 11 present the performance evaluation metrics for datasets from both the locations for all the model variants and horizons. The MSE, RMSE, R^2 are indicated, with the best performing model also denoted in bold in all the tables. The MSE, RMSE, and R^2 score are multiplied by 10^3, 10^2, and 10^2 respectively for ease of representation.

It can be observed that the proposed model outperforms all the other models in both cases. The least values for the performance evaluation metrics of MSE, RMSE, and the maximum values for the R^2 score can be observed for the proposed model.

For a horizon length of 1 day ahead forecasting, in location 1, it can be observed that the MSE value improves as the model is introduced with the additional components of attention, convolution and bidirectional LSTM. Similarly, in location 2, the forecast performance in terms of MSE improves with the additional gates of LSTM. The RMSE providing an insight into the error also follows similar trends with MSE. The R^2 score being the coefficient of determination reflects better performance when it is close to 1. Here, since we have multiplied the values with 10^2, the R^2 scores close to 100 are better as compared to others. This score can be observed to be highest for the proposed model for both the locations. The variants of RNN were introduced to tackle the vanishing and exploding gradient problem. Now, LSTM, with its complex gating mechanism, is capable of handling long-term dependencies. The attention mechanism again further enhances this.

Now, for a horizon length of 4, after the proposed model, the LSTM shows better performance for both locations. It can be observed in terms of MSE, RMSE and R^2 score.

Table 10 Performance Evaluation for 4 Horizon on Location 2

Model	MSE	RMSE	R^2
RNN	11.102	10.5369	61.1225
LSTM	10.638	10.3141	62.7496
GRU	10.562	10.2773	63.0147
Bidir-LSTM	10.762	10.3741	62.3147
CNN-LSTM	10.624	10.3074	62.7979
Attention LSTM	10.608	10.2997	62.8534
Proposed	**10.444**	**10.2198**	**63.4276**

Table 11 Performance Evaluation for 10 Horizon on Location 2

Model	MSE	RMSE	R^2
RNN	12.577	11.2151	56.1274
LSTM	12.453	11.1593	56.5629
GRU	12.423	11.1459	56.6670
Bidir-LSTM	12.596	11.2233	56.0628
CNN-LSTM	12.335	11.1066	56.9720
Attention LSTM	12.366	11.1203	56.8660
Proposed	**12.325**	**11.1022**	**57.0063**

In the case of 10 days ahead forecasting for location 1, the Bidirectional LSTM, the CNN LSTM and attention LSTM tend to give more accurate results after the proposed model. For location 2, with 10 days horizon, the proposed model is followed by the CNN LSTM and the attention LSTM. The MSE, RMSE, and R^2 metrics of performance evaluation tend to indicate similar trends in reflecting the model accuracy in forecasting future data.

The results are shown for the best performing optimizer in all the cases. Since no single optimizer can be called the best, experiments were performed on six different optimizer types. Further analysis is shown in the sections ahead. Also, an additional learning rate tuning mechanism of warm restarts was tested, resulting in better performance in terms of MSE, RMSE, and R^2 score. The bias and variance scores could further highlight the importance of this hyperparameter tuning mechanism.

The proposed model is an enhancement to the existing models implemented here since it introduces an attention-based architecture with the CNN and LSTM components for solar irradiance forecasting. Also, it integrates the learning rate tuning mechanism of warm restarts along with the optimal optimizer chosen from six different optimizers. Additionally, the bias-variance components of MSE decomposition add to the further robustness of the model. This is further discussed in the next section. The performance metrics for both locations indicate the superiority of the proposed model. Even when the length of the horizon increases, the proposed model provides the optimal performance in all the cases. The RNN, GRU, and LSTM show low performance in most cases while other models perform better. The developed attention LSTM and CNN LSTM also perform at par with the proposed model. The observations prove that the CNN works well with the LSTM architecture when forecasting solar data, and the attention mechanism adds to the robustness of the model.

5.2 Comparison of bias-variance

Figure 9 indicates the bias-variance changes according to the change in models and horizons. Bias values are normalized by multiplying the values with 10^2 for all the horizons. In the case of variance, it is multiplied by $E - 15$ for proper visualization. The Fig. 9 represent how bias-variance can also be used as the deciding mechanism and the conventional performance evaluation metrics for finalizing the model to attain the best possible results.

The figures indicate the performance in terms of bias and variance. The results are similar to the ones obtained for other standard performance evaluation metrics. Utilizing these bias-variance metrics, the tuning and optimizing of the model was also performed to attain an accurate forecast setting.

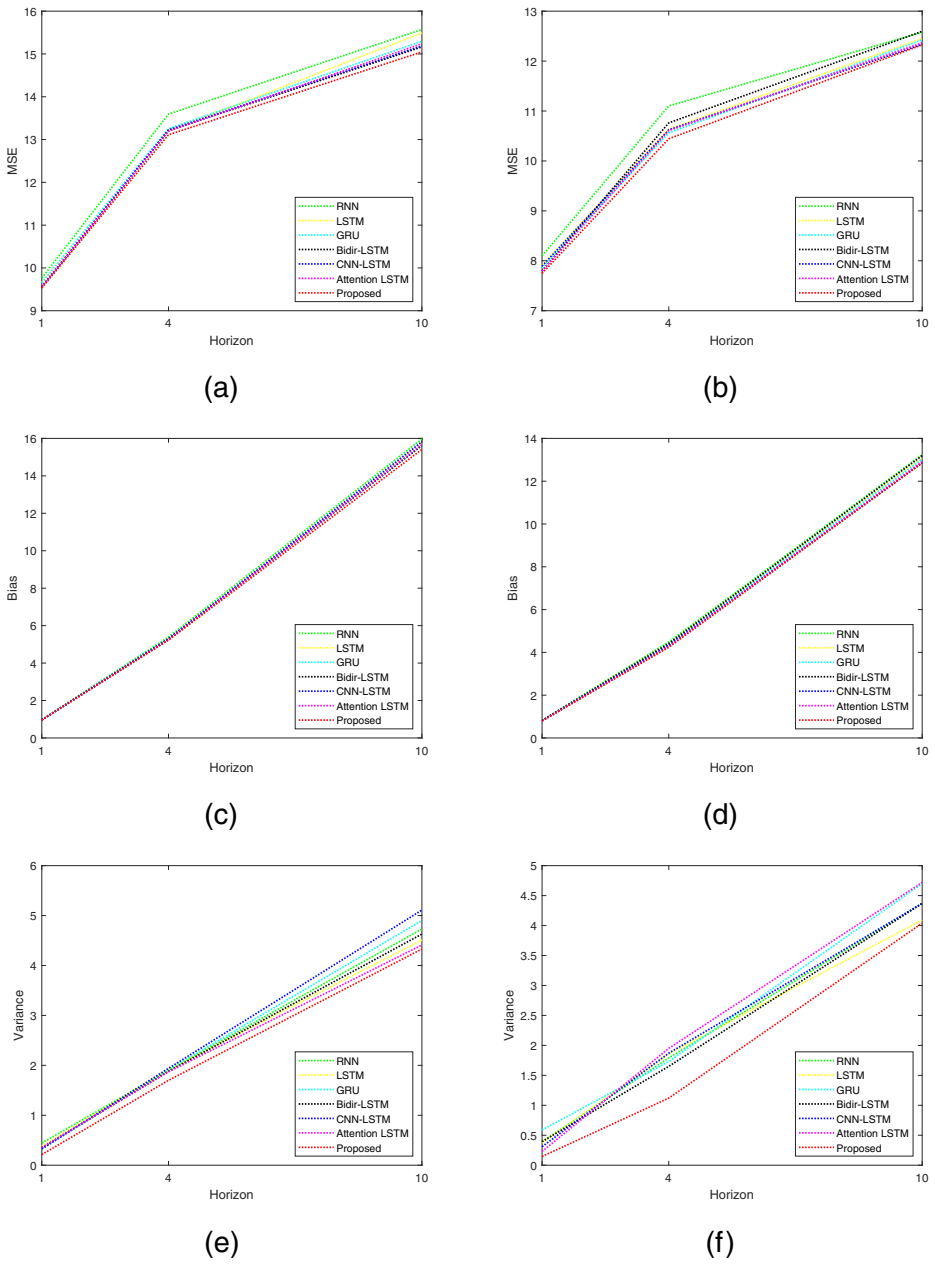

Fig. 9 Error metrics for both the locations corresponding to horizon length: (**a**) MSE for Location 1, (**b**) MSE for Location 2, (**c**) Bias for Location 1, (**d**) Bias for Location 2, (**e**) Variance for Location 1, (**f**) Variance for Location 2

Table 12 Performance Evaluation for 1 Horizon

Model	Location 1			Location 2		
	MSE	RMSE	R^2	MSE	RMSE	R^2
AdaGrad	9.822	9.9110	68.2925	8.158	9.0322	71.4341
Adadelta	14.762	12.1501	52.3472	12.340	11.1087	56.7894
Adam	9.554	9.7748	69.1576	7.800	8.8322	72.6850
AdaMax	9.531	9.7627	69.2339	7.747	8.8020	72.8714
Nadam	9.570	9.7827	69.1081	7.765	8.8123	72.8077
AMSGrad	9.564	9.7797	69.1272	7.769	8.8144	72.7949

Bias measures the error of the forecasting model in learning the underlying patterns in a time series. Variance measures the sensitivity of the estimation function to the training dataset. The proposed model reflects the least bias and variance. Following the framework of bias-variance calculation, the hyperparameters are also tuned accordingly. The learning rate tuning mechanism of warm restarts reflects lower bias and variance, which indicates that it can be utilized for solar irradiance forecasting.

The results obtained help us gain further insight into the model error components when building a deep neural network model for time series forecasting. The reduction of these components can further improve the learning and generalization performance of the developed model.

5.3 Comparison of optimizers

Evaluation of the optimizer variants resulted in the following observations. Tables 12, 13 and 14 indicate the performance of all the six optimizers on both the datasets for the three horizons. The SGD variant with the Nesterov momentum variant was evaluated as well, but it took longer epochs to train and generalize, which is not always feasible in the presence of optimizers such as ADAM.

Firstly, all the optimizer variants perform well in the solar irradiance forecast scenario. The least performance could be observed for the Adadelta optimizer. In some cases, ADAM performs the best while in some AdaMax or the AMSGrad variant. There is not one optimizer performing universally the best. However, the ADAMAX variant performs optimally

Table 13 Performance Evaluation for 4 Horizon

Model	Location 1			Location 2		
	MSE	RMSE	R^2	MSE	RMSE	R^2
AdaGrad	13.487	11.6134	56.4646	10.800	10.3926	62.1801
Adadelta	15.855	12.5917	48.8210	13.097	11.4445	54.1363
Adam	13.179	11.4803	57.4569	10.461	10.2283	63.3665
AdaMax	13.103	11.4469	57.7038	10.444	10.2198	63.4276
Nadam	13.234	11.5040	57.2813	10.532	10.2628	63.1189
AMSGrad	13.111	11.4504	57.6784	10.511	10.2525	63.1933

Table 14 Performance Evaluation for 10 Horizon

Model	Location 1			Location 2		
	MSE	RMSE	R^2	MSE	RMSE	R^2
AdaGrad	15.695	12.5280	49.5843	12.6178	11.2329	55.9880
Adadelta	19.690	14.0321	36.7492	16.7002	12.9229	41.7479
Adam	15.184	12.3223	51.2269	12.4109	11.1404	56.7096
AdaMax	15.324	12.3791	50.7761	12.4100	11.1400	56.7128
Nadam	15.115	12.2946	51.4460	12.5252	11.1916	56.3108
AMSGrad	15.143	12.3060	51.3558	12.3357	11.1066	56.9720

faster while the others training takes a longer duration of time. The Nadam variant also performs optimally.

Some of the optimizers reach minima quickly while others take time. Also, the weight initialization can lead to randomness. Any single optimizer cannot be called the best. However, fine-tuning can lead to higher accuracy. It can be observed that Adam has close results in most of the configurations and is suitable for large datasets and high dimensional space. Its variants of AdaMax and AMSGrad generally improve the results.

5.4 Comparison of horizons

The performance of the models is also shown for three forecast horizons. As the horizon length increases, the model performance differs in a trend similar for all the scenarios. Figures 10, 11 and 12 show the performance change of the proposed model as the horizon increases for both the locations. The MSE, Bias, and Variance show better performance metrics in the case of short horizons.

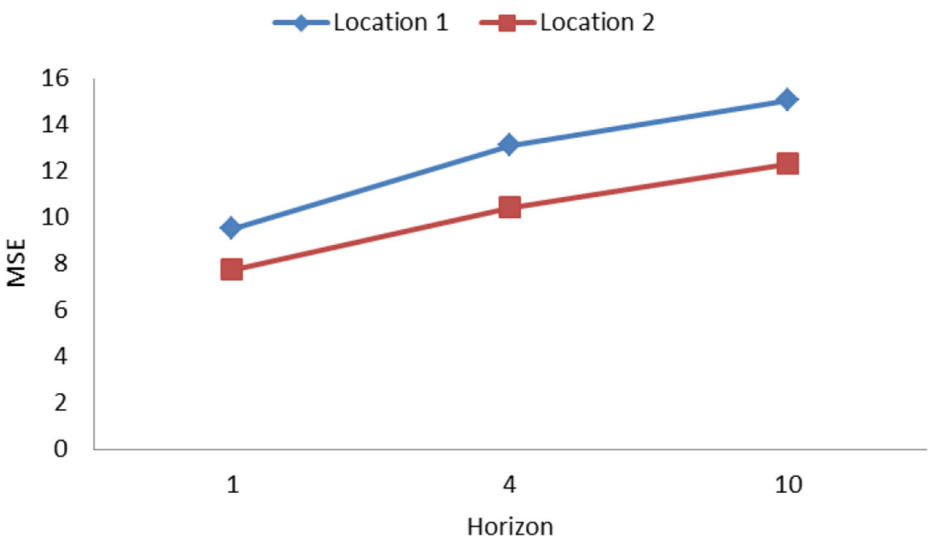

Fig. 10 Forecast Horizon vs MSE

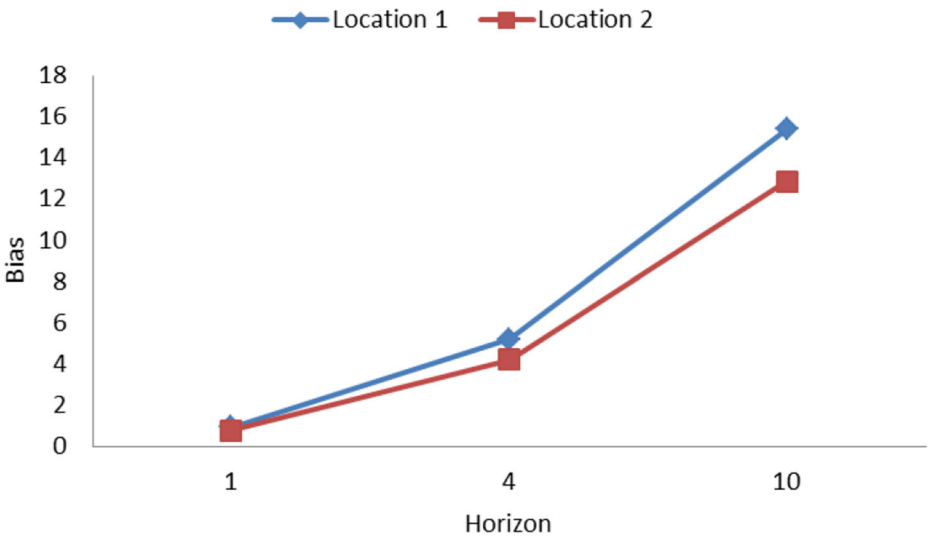

Fig. 11 Forecast Horizon vs Bias

Model performance improves when the learning rate is tuned using the warm restarts method. The evaluation metrics showed better figures indicating LR tuning with warm restarts helps in the model's generalization when used with ADAM and its variants. MSE gives a rough overview about which model variant might be better performing, but bias-variance helps finalize the one model that performs the best. This can be seen in the tabular analysis observations, proving that MSE, along with the model's bias and variance scores, can aid in generalization. Model performance improves when the proposed method is used for forecasting. The evaluation metrics show better figures with LR tuning with warm restarts, indicating better model generalization when used with ADAM and its variants.

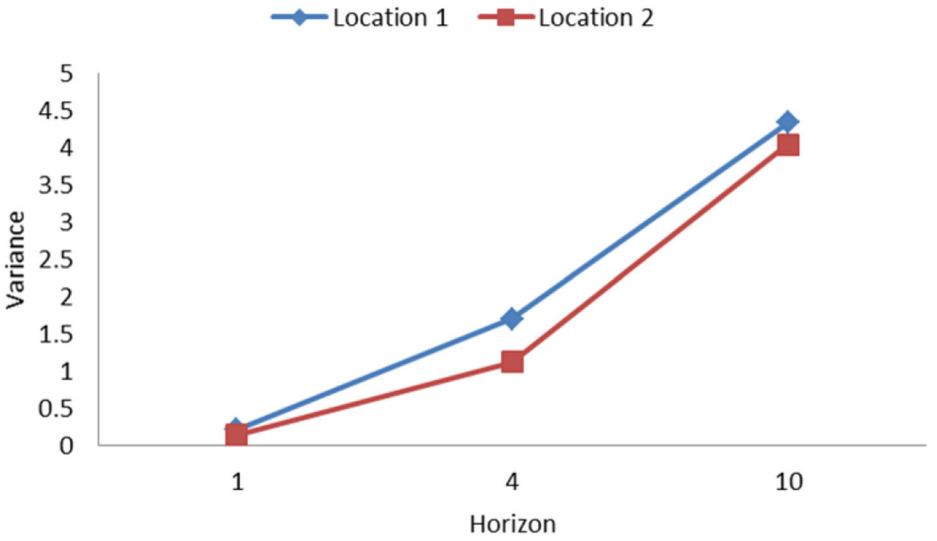

Fig. 12 Forecast Horizon vs Variance

Then again, model selection and tuning is a crucial step in model development. Without this, a good model might be considered a weaker one.

Conclusively, the deep learning solar irradiance forecast mechanisms were discussed, and their results were shown. The proposed methodology for forecasting introduces bias-variance analysis as a crucial step in the model building process. This was proven through experimentations with varied models, optimizers, and parameters finalizing upon the best forecast combination with the proposed methodology's help. The proposed CNN-Attention LSTM architecture proves to be a superior model capable of reliable solar irradiance forecasting.

6 Conclusion

In this work, the solar irradiance data is analyzed from NASA's ArcGIS interface as a visual system. The data obtained in the form of spatial information provides insight into the region's solar irradiance scenario. A novel deep neural network forecast model called the CNN–Attention LSTM is proposed here for solar irradiance forecasting. The proposed model was evaluated against many deep learning forecasting models. The model reported an optimal forecast performance with the lowest MSE and RMSE, outperforming other time series models. The analysis indicated that the LSTM models are an efficient model for solar time series data. Its utilization with additional attention mechanism and convolutional layers provides a significant boost to increase the forecasting performance. Also, the proposed modeling framework for forecasting can be easily utilized for extended scientific areas of predicting in areas such as renewable energy, wind speed, weather data, etc., using the ArcGIS data without many constraints. The time series data's internal representations and temporal dependencies could be well exploited through the utilization of convolutional layers and attention LSTM mechanism.

Acknowledgements The data used in the research were obtained from the NASA Langley Research Center (LaRC) POWER Project funded through the NASA Earth Science/Applied Science Program. The dataset is available at the website https://power.larc.nasa.gov/data-access-viewer/. We acknowledge to Madhya Pradesh Council of Science and Technology, Bhopal, India for research support.

Declarations

Conflict of Interests The authors declare that they have no conflict of interest.

References

1. Aburto L, Weber R (2007) Improved supply chain management based on hybrid demand forecasts. Applied Soft Computing Journal 7(1):136–144. https://doi.org/10.1016/j.asoc.2005.06.001
2. Alex G, Greg W, Ivo D (2014) Neural turing machines
3. Amrouche B, Le Pivert X (2014) Artificial neural network based daily local forecasting for global solar radiation. Appl Energy 130:333–341. https://doi.org/10.1016/j.apenergy.2014.05.055
4. Amrouche B, Sicot L, Guessoum A, Belhamel M (2013) Experimental analysis of the maximum power point's properties for four photovoltaic modules from different technologies: Monocrystalline and polycrystalline silicon, CIS and CdTe, vol 118
5. Bahdanau D, Cho K, Bengio Y (2015) Neural machine translation by jointly learning to align and translate. CoRR abs/1409.0473

6. Berardi VL, Zhang GP (2003) An empirical investigation of bias and variance in time series forecasting: Modeling considerations and error evaluation. IEEE Transactions on Neural Networks 14(3):668–679. https://doi.org/10.1109/TNN.2003.810601
7. Box GEP, Jenkins GM, Reinsel GC, Ljung GM (2015) Time Series Analysis: Forecasting & Control
8. Brahma B, Wadhvani R (2020) Solar irradiance forecasting based on deep learning methodologies and multi-site data. Symmetry, 12(11). https://doi.org/10.3390/sym12111830, https://www.mdpi.com/2073-8994/12/11/1830
9. Brahma B, Wadhvani R (2020) Time series forecasting: A comparison of deepneural network techniques. Solid State Technology 63:1747–1761. http://solidstatetechnology.us/index.php/JSST/article/view/2404
10. Brahma B, Wadhvani R, Shukla S (0) Attention mechanism for developing wind speed and solar irradiance forecasting models. Wind Eng 0(0):0309524X20981885. https://doi.org/10.1177/0309524X20981885
11. Brockwell PJ, Davis RA (2002) Introduction to Time Series and Forecasting - Second Edition
12. Creal D, Koopman SJ, Lucas A (2013) Generalized autoregressive score models with applications. J Appl Econom 28(5):777–795. https://doi.org/10.1002/jae.1279
13. Dong N, Chang J-F, Wu A-G, Gao Z-K (2020) A novel convolutional neural network framework based solar irradiance prediction method. International Journal of Electrical Power and Energy Systems 114:105411. https://doi.org/10.1016/j.ijepes.2019.105411. http://www.sciencedirect.com/science/article/pii/S0142061518332915
14. Dozat T (2016) Incorporating nesterov momentum into adam
15. Duchi J, Hazan E, Singer Y (2011) Adaptive subgradient methods for online learning and stochastic optimization. J Mach Learn Res 12(61):2121–2159. http://jmlr.org/papers/v12/duchi11a.html
16. Geron A (2017) Hands-on machine learning with Scikit-Learn and TensorFlow : concepts, tools, and techniques to build intelligent systems
17. Graves A, Schmidhuber J (2005) Framewise phoneme classification with bidirectional LSTM and other neural network architectures. In: Neural Networks, 18, pp 602–610
18. Greff K, Srivastava RK, Koutnik J, Steunebrink B, Schmidhuber J (2017) LSTM: A Search Space Odyssey. IEEE Transactions on Neural Networks and Learning Systems 28(10):2222–2232. https://doi.org/10.1109/TNNLS.2016.2582924
19. Guermoui M, Melgani F, Gairaa K, Mekhalfi ML (2020) A comprehensive review of hybrid models for solar radiation forecasting. J Clean Prod 258:120357. https://doi.org/10.1016/j.jclepro.2020.120357. http://www.sciencedirect.com/science/article/pii/S0959652620304042
20. Heng J, Wang J, Xiao L, Lu H (2017) Research and application of a combined model based on frequent pattern growth algorithm and multi-objective optimization for solar radiation forecasting. Appl Energy 208:845–866. https://doi.org/10.1016/j.apenergy.2017.09.063
21. 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
22. hsiang Wang C, Grozev G, Seo S (2012) Decomposition and statistical analysis for regional electricity demand forecasting. Energy 41(1):313–325. https://doi.org/10.1016/j.energy.2012.03.011
23. Ilya S, Oriol V, Quoc VL (2014) Sequence to sequence learning with neural networks
24. Kariniotakis G (2017) Renewable energy forecasting: From models to applications
25. Kim TY, Cho SB (2019) Predicting residential energy consumption using CNN-LSTM neural networks. Energy 182:72–81. https://doi.org/10.1016/j.energy.2019.05.230
26. Kingma DP, Ba JL (2015) Adam: A method for stochastic gradient descent. ICLR: International Conference on Learning Representations
27. Kyunghyun C, Bart VM, Caglar G, Dzmitry B, Fethi B, Holger S, Yoshua B (2014) Learning phrase representations using rnn encoder-decoder for statistical machine translation
28. LeCun Y, Haffner P, Bottou L, Bengio Y Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)
29. Li T, Hua M, Wu X (2020) A Hybrid CNN-LSTM Model for Forecasting Particulate Matter (PM2.5). IEEE Access 8:26933–26940. https://doi.org/10.1109/ACCESS.2020.2971348
30. Li Y, Su Y, Shu L (2014) An ARMAX model for forecasting the power output of a grid connected photovoltaic system. Renew Energy 66:78–89. https://doi.org/10.1016/j.renene.2013.11.067
31. Loshchilov I, Hutter F (2016) SGDR: stochastic gradient descent with restarts. CoRR abs/1608.03983, http://arxiv.org/abs/1608.03983
32. Lubitz WD (2011) Effect of manual tilt adjustments on incident irradiance on fixed and tracking solar panels. Appl Energy 88(5):1710–1719. https://doi.org/10.1016/j.apenergy.2010.11.008
33. Mateo F, Carrasco JJe, Sellami A, Millán-Giraldo M, Domínguez M, Soria-Olivas E (2013) Machine learning methods to forecast temperature in buildings. Expert Syst Appl 40(4):1061–1068. https://doi.org/10.1016/j.eswa.2012.08.030

34. Neves C, Fernandes C, Hoeltgebaum H (2017) Five different distributions for the Lee-Carter model of mortality forecasting: A comparison using GAS models. Insurance: Mathematics and Economics 75:48–57. https://doi.org/10.1016/j.insmatheco.2017.04.004
35. 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
36. Reddi SJ, Kale S, Kumar S (2018) On the convergence of Adam and beyond. In: 6th International Conference on Learning Representations, ICLR 2018 - Conference Track Proceedings
37. Rumelhart DE, Hinton GE, Williams RJ (1986) Learning representations by back-propagating errors. Nature 323(6088):533–536. https://doi.org/10.1038/323533a0
38. Shih SY, Sun FK, yi Lee H (2019) Temporal pattern attention for multivariate time series forecasting. Mach Learn 108(8-9):1421–1441. https://doi.org/10.1007/s10994-019-05815-0
39. Srivastava S, Lessmann S (2018) A comparative study of LSTM neural networks in forecasting day-ahead global horizontal irradiance with satellite data. Sol Energy 162:232–247. https://doi.org/10.1016/j.solener.2018.01.005
40. Taieb SB, Atiya AF (2016) A Bias and Variance Analysis for Multistep-Ahead Time Series Forecasting. IEEE Transactions on Neural Networks and Learning Systems 27(1):62–76. https://doi.org/10.1109/TNNLS.2015.2411629
41. Tsay RS (2005) Analysis of Financial Time Series Second Edition
42. Wang Y, Shen Y, Mao S, Chen X, Zou H (2018) LASSO & LSTM Integrated Temporal Model for Short-term Solar Intensity Forecasting. IEEE Internet Things J., https://doi.org/10.1109/JIOT.2018.2877510
43. Yu XH, Chen GA, Cheng SX (1995) Dynamic Learning Rate Optimization of the Backpropagation Algorithm. IEEE Transactions on Neural Networks 6(3):669–677. https://doi.org/10.1109/72.377972
44. Zang H, Liu L, Sun L, Cheng L, Wei Z, Sun G (2020) Short-term global horizontal irradiance forecasting based on a hybrid cnn-lstm model with spatiotemporal correlations. Renew Energy 160:26–41. https://doi.org/10.1016/j.renene.2020.05.150. http://www.sciencedirect.com/science/article/pii/S0960148120308557
45. Zeiler MD (2012) ADADELTA: an adaptive learning rate method. CoRR abs/1212.5701, http://arxiv.org/abs/1212.5701
46. Zhang PG (2003) Time series forecasting using a hybrid ARIMA and neural network model. Neurocomputing 50:159–175. https://doi.org/10.1016/S0925-2312(01)00702-0