Real-time solar power forecasting using LSTM algorithms

Leelavathi M[a] and Suresh Kumar V[b]
[a,b]Thiagarajar College of Engineering, Madurai, Tamil Nadu, India

Abstract

Renewable energy sources, particularly solar energy systems, have an adequate power supply (PS). But still, PS is widely dispersed and heavily weather-dependent. Hence, forecasting solar output offers a robust answer to such issues. Also, it reduces the workload of delivering electricity through transmission and distribution. In this study, deep learning algorithms are used for precise solar power forecasting (SPF), utilizing a short-term forecasting method. In particular, a long short-term memory (LSTM) approach is used for SPF in the time-series dataset. This paper aims to prevent over-fitting issues that arise in time-series datasets. The real-time solar power time-series dataset is collected from Thiagarajar College of Engineering, Madurai. The SPF are used to evaluate the accuracy using different error types, including mean absolute error(MAE), mean squared error(MSE), and root mean square error(RMSE). Additionally, other algorithms like AdaBoost, linear regression, artificial neural networks, neural networks, space vector machines, and recurrent neural networks are compared to the proposed LSTM algorithm.

Keywords: solar power forecasting, artificial intelligence algorithms, renewable energy, deep learning algorithms, machine learning algorithms

1. Introduction

The demand for electricity increases as the population grows. To meet demand, more electricity must be produced. One of the most important ways to meet the demand for electricity is by using renewable energy, particularly solar energy. Globally, grid connections are being integrated more and more with solar energy sources. To meet the power demand, the frequent forecasting method is one of the best solutions. Additionally, inconsistent, erratic, and variable energy production limits the development of solar energy [1]. As a result, reducing these limits by forecasting solar energy is one of the best solutions.

Several types of calculations can be used to produce accurate forecasts. These include mathematical, artificial intelligence, physical, and hybrid forecasting methods. The four forecasting methods are reviewed in the following. Physical methods can be used in a variety of ways, including numerical local weather forecasts, ground-based sensor measurements, images of the entire sky, and satellite imagery. Physical methods [2] are based on these inputs and should be stable, like a clear sky, while forecasting is inaccurate under cloudy conditions.

Additionally, there are various mathematical methods [3], such as regression, exponential smoothing, multivariate ARMA (ARMAX), and auto-regressive moving averages (ARMAs). It is typically used in conjunction with mathematical specifications and historical information. However, these approaches frequently rely on linear predictions [6], which may require accuracy in the nonlinear mapping between solar inputs and outputs.

Forecasting algorithms are developed using various AI methods [4], such as meta-heuristic algorithms, machine learning, deep learning algorithms, and space vector machines. This mathematical concept is applied to various domains, including forecasting. As a result of pre-processing, training, and testing, the algorithm generates precise outputs. By combining two or more effective methods, a hybrid strategy achieves a result. Several typical instances of hybrid-method forecasting are provided in [5].

As a result, artificial intelligence (AI) technology produces more precise forecasts than any other technology, particularly when employing deep learning (DL) algorithms [6], which are more commonly employed for forecasting. Convolutional neural networks, long-short-term memory (LSTM) networks [7], and recurrent neural networks are only a few of the types that the DL algorithm describes. The LSTM network, which is based on a neural network method with several hidden layers and a modified architecture, also provides convenient nonlinear fitting of arbitrary situations. A high-precision LSTM algorithm for categorizing weather types using k-means and forecasting PV output power using a clearness index approach was presented. Even on foggy days, this method was created to improve prediction accuracy.

In this study, the day-ahead solar power forecast is forecast using the LSTM algorithm, which uses sparse training data to create precise forecasts with positive outcomes. This paper provides comprehensive details on the LSTM algorithm training procedure.

2. Data summary

The raw data for this study was collected from a 1 kW photovoltaic system at the Thiagarajar College of Engineering in Madurai. The site information for the 1KV PV system is displayed in Table I. The solution incorporates a cloud-based mechanism to keep track of the initially gathered data. Based on time-series data, raw data is collected every second from 00:00 to 23:59. The raw data shows some missing or outlier data as a result. During data pre-processing, the interpolation approach fills in missing values by using eq.1. Figure 1 depicts the results of the data analysis after preprocessing.

$$LI = (E_v - S_v)/(M_n + 1) \tag{1}$$

where
LI= Linear interpolation
E_v = End value
S_v = Starting value
M_n = Number of missing value count

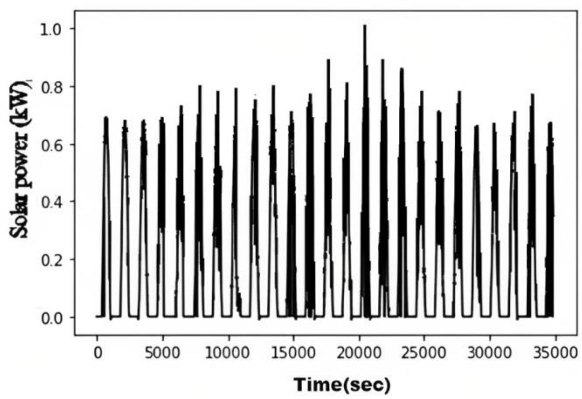

Fig. 1. Solar power (kW) experimental data set

2.1. Model validation

The algorithm is tested using 70% of the collected raw data, 15% of the trained data, and 5% of the verified data. Eqs. 2, 3, and 4 evaluate prediction accuracy using the mean square error (MSE), mean absolute error (MAE), and root means square error (RMSE). The difference between the actual and the predicted values are calculated using MSE. The deviation between the observed and the projected value is referred to as the average MAE. When the value is lower, the difference is negligible. The dispersion between expected and actual values is measured by the RMSE. These three evaluation measures can be used to judge a model's accuracy.

$$MSE = \frac{1}{m}\sum_{p=1}^{m}(S_i - S_o) \tag{2}$$

$$MAE = \sum_{p=1}^{m}\frac{|S_i - S_o|}{m} \tag{3}$$

$$RMSE = \sqrt{\sum_{p=1}^{m}\frac{(S_i - S_o)^2}{m}} \tag{4}$$

Where,
p = variable of p

m = Total solar power
Si = Actual solar power
So = Predicted solar power

3. Proposed methodology

This section covers LSTM algorithm training based on solar output predictions.

3.1. Long short-term memory

A feature of the LSTM architecture is the ability to read, write, or save data in the previous state, as described in Fig. 2. It performs like a memory. During training, the information is either passed or blocked based on the weight of the input. The weight is comparable to the data that is kept in a cell. Basic parameters for the LSTM include the input vector, the output vector, the forget gate, and the cell state.

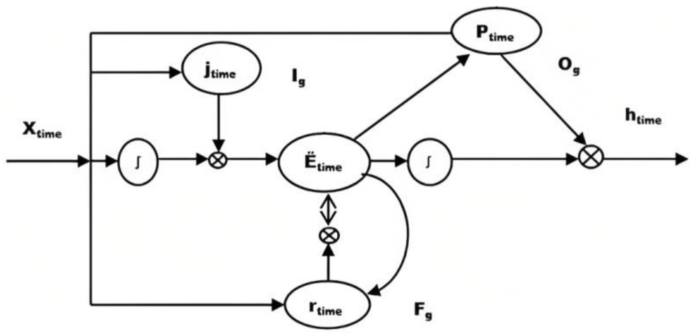

Fig.2. LSTM architecture

The LSTM is provided with an input vector (I_g) at time instance (j_{time}) and its output vector (o_g) at time instance h_{time-1}. To manage the network, LSTM employs several gates. Both Equation (1) and Fig. 2 depict the forget gate in action. The forget gate's current input is X_{time}, and its prior output is h_{time-1} in Eq. (1). The decision of whether the gate should remember or forget earlier information is made using sigmoid functions. It is noteworthy that both the bias vector b_r and the weight matrix W_r are used.

$$r_{time} = \sigma(W_r * [h_{time-1}, x_{time}] + b_r) \tag{5}$$

where

σ: Sigmoid activation function

W_r is a Weight and B_r is a bias for forget vector

x_{time} : Input in time

h_{time-1} : Output hidden layer in time-1

Figure 3 depicts an LSTM cell. In that case, the input gate (I_g) regulates the amount of current data sent to the newly created cell. A sigmoid function is used to calculate the updated value, as shown in Eq. (6). A new value, $\ddot{E}_{time}$ is also generated by the tanh function in Eq. (7), which will be added to the state value.

$$j_{time} = \sigma(W_j * [h_{time-1}, x_{time}] + b_j) \tag{6}$$

$$\ddot{E}_{time} = tanh(W_{\ddot{E}} * [h_{time-1}, x_{time}] + b_{\ddot{E}}) \tag{7}$$

where

W_j is a Weight and B_j is a bias for input vector

$W_{\ddot{E}}$ is a Weight and $B_{\ddot{E}}$ is a bias for cell vector

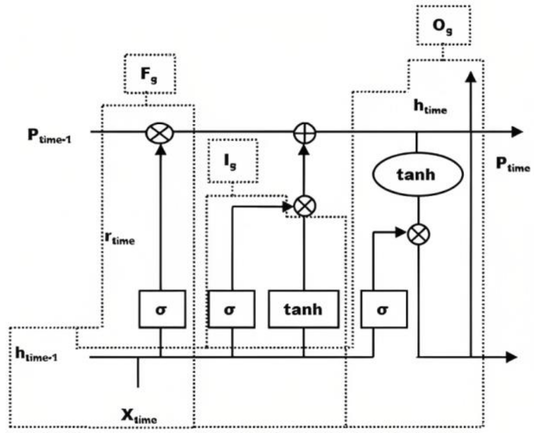

Fig. 3. LSTM cells.

The new cell vector E_{time} is made by increasing the output of the forget vector r_{time} by the old cell vector E_{time-1}, which is then added to it as $\ddot{E}_{time}$. The cell status affects the resultant h_{time}, despite filtering.

$$E_{time} = r_{time} * E_{time-1} + j_{time} * \ddot{E}_{time} \tag{8}$$

As shown in Eqs. (9) and (10), the new cell state E_{time} is increased by the result using a sigmoid function, P_{time} after passing through a tanh function.

$$P_{time} = \sigma\,(W_P * [h_{time-1}\,, x_{time}] + b_P) \tag{9}$$

$$h_{time} = P_{time} + tanh\,(E_{time}) \tag{10}$$

where

W_p is a weight and B_p is a bias for output vector

tanh : tangent hyperbolic

3.2. Training process for LSTM

The current data is determined by the forget gate. Input gates determine step states using the current data of the forget gates. The output gate selects the following hidden state at the end. The training process must be used to learn the weight and bias parameters. The bias ranges from [-1 to 1], while the weight has values between 0 and 1. The LSTM networks are trained using a variety of supervised learning approaches.

The LSTM must be calculated using gradient descent and back-propagation over time. The following cell states are applied to h_{time} gradients, E_{time-1}, and the cell state E_{time} during the backward pass. A suitable stochastic gradient descent solution is then used to update the weights. The accurate output is displayed at the output layer based on the revised weights.

The proposed system structure for this investigation is simulated in Fig. 4. In that case, the input is experimentally collected raw data that is first produced as initial pre-processing to correct the raw data. Then the LSTM training process is executed, and precise forecasts are produced. Considering that the precise output values are analyzed using different error and accuracy levels,

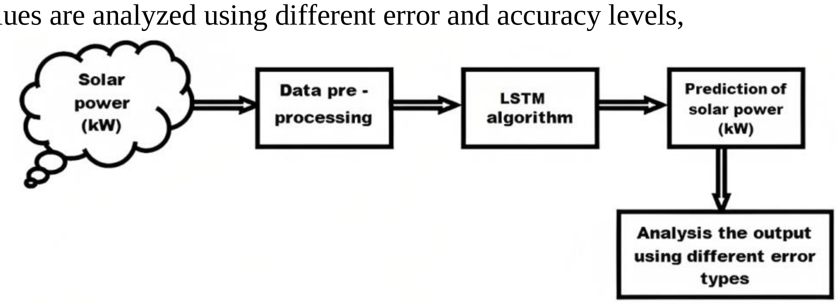

Fig. 4. Proposed system framework

4. Experimental result and discussion

This section describes a performance evaluation and analysis of the proposed algorithm.

4.1. Analysis of prediction results

The results of the prediction accuracy of different AI algorithms are shown in Table 1. The LR's respective MSE, MAE, and RMSE error values are 0.0520, 0.2000, and 0.2270. Then, for NN, the MSE, MAE, and RMSE error values are each 0.0530, 0.2270, and 0.2000, respectively. The ANN's MSE, MAE, and RMSE error values are, respectively, 0.0080, 0.0060, and 0.0113. The MSE, MAE, and RMSE error values for SVM are then 0.0520, 0.2000, and 0.2100, respectively. The RNN's MSE, MAE, and RMSE error values are, respectively, 0.0774, 0.0098, and 0.1421. The MSE, MAE, and RMSE error values for Ada Boost are 0.0010, 0.0100, and 0.032, respectively. The proposed algorithm with the lowest error value and maximum accuracy has 96% accuracy with an MSE of $1.5089e^{-04}$, an MAE of 0.0020, and an RMSE of 0.0077. Compared to the other algorithms, the proposed LSTM method performed better in this study.

Table 1. Compared and contrasted various AI algorithms for obtaining accurate forecasting results.

Sl. no.	Prediction algorithm	MSE	MAE	RMSE	Accuracy %
1.	Linear regression	0.0520	0.2000	0.2270	81.8271
2.	NN	0.0530	0.2270	0.2000	95.8
3.	ANN	0.0080	0.0060	0.0113	93.9896
4.	SVM	0.0520	0.2000	0.2000	97.5195
5.	RNN	0.0774	0.0098	0.1421	94.22
6.	AdaBoost	0.0100	0.0100	0.0100	95.88
7.	LSTM	$1.5089e^{-04}$	0.0020	0.0077	98.2392

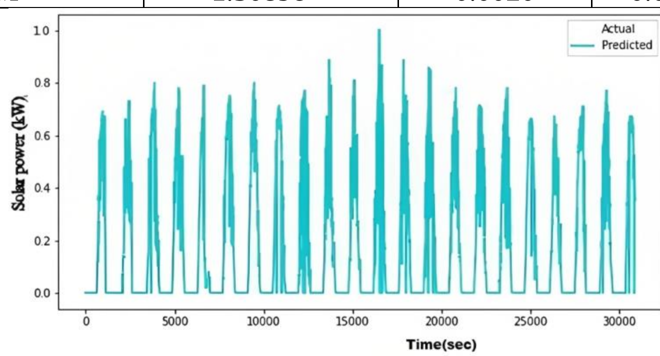

Fig. 5. Actual and predicted forecasting results for solar power (kW) using the LSTM algorithm

4.2. Discussion of the result

The actual data is displayed based on the sunrise and sunset. The actual data are trained with different AI algorithms. Based on the trained data, some samples of 20 data points are shown in Fig. 4. In that case, the SVM, LR, and NN are over-trained (i.e., an over fitting problem occurred). Then AdaBoost and RF are more or less trained on similar data with the same inaccuracy of prediction value. Then, the ANN and RNN also produced the over fitting problem with the trained data. Finally, the proposed LSTM-trained data are accurate data based on the actual dataset. Because using the stochastic gradient descent updates the correct weight values.

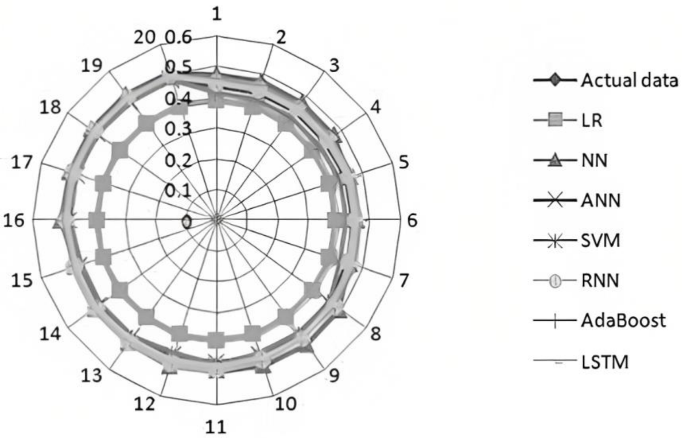

Fig. 6. Spider graph for SPF in different AI algorithms

However, LSTM algorithms provide a robust evaluation of solar power forecasting using sparse data. According to their expertise in PV, the results demonstrate LSTM's efficiency over other algorithms. LSTM was trained based on observations and the principles of neural networks. As a result, the predicted performance of LSTM can be greatly improved. The training of sparse data for prediction should produce an accurate result with less error. From the overall result of this research work, the proposed LSTM reduced the over fitting problems and produced accurate output values.

5. Conclusion

The proposed LSTM algorithm is used in the short-term approach to forecasting solar power. The precise outputs are analyzed using the different error types and accuracy. For the precise output in the proposed LSTM algorithm, there is a minimum error value where the MSE is $1.5089e^{-04}$, the MAE is 0.0020, and the RMSE is 0.0077. The proposed algorithm is then contrasted with other algorithms using various AI technologies. The other AI algorithms included LR, NN, ANN, RNN, and Ada Boosting. The proposed algorithm achieved 96% accuracy when contrasted to other AI algorithms. Future research should examine data from various weather conditions.

6. Reference

1. Li SJ, Gong WY, Yan XS, et al. Parameter extraction of photovoltaic models using an improved teaching-learning-based optimization. Energy Convers Manage 2019; 186:293–305.
2. Hocaoˇglu, Fatih O and Gerek, Omer N, and Kurban, Mehmet(2008), Hourly solar radiation forecasting using optimal coefficient 2-D linear filters and feed-forward neural networks, Solar energy, 82(8): pp. 714-726.
3. Chaturvedi, DK and Isha, I(2016), Solar power forecasting: A review, International Journal of Computer Applications,145(6): pp. 28-50.
4. N. Amjady, F. Keynia, and H. Zareipour, "Wind power prediction by a new forecast engine composed of modified hybrid neural network and enhanced particle swarm optimization," *IEEE Trans. Sustain. Energy*, vol. 2, no. 3, pp. 265–276, Jul. 2011
5. Reikard, Gordon(2009), Predicting solar radiation at high resolutions: A comparison of time series forecasts, Solar energy,83(3): pp. 342-349.
6. Li, Gangqiang, et al. "Photovoltaic power forecasting with a hybrid deep learning approach." IEEE Access 8 (2020): 175871-175880.
7. Wang, Huaizhi, et al. "A review of deep learning for renewable energy forecasting." Energy Conversion and Management 198 (2019): 111799.
8. Liu, Chun-Hung, Jyh-Cherng Gu, and Ming-Ta Yang. "A simplified LSTM neural networks for one day-ahead solar power forecasting." IEEE Access 9 (2021): 17174-17195.