



Integrating LSTM and Flutter For Real-Time Solar DC Power Prediction

Chigilipalli Bharat Kumar

Department of Computer Science and Systems Engineering, Andhra University College of Engineering (A) ,
Andhra University , Visakhapatnam - 530003, Andhra Pradesh, India

* Corresponding Author : Chigilipalli Bharat Kumar ; bharat@vignaniit.edu.in

Abstract: Renewable energy sources, particularly solar energy, are essential to meeting the world's growing energy needs. Real-time prediction of DC power generation is necessary for optimizing energy management and utilization of solar energy. We initially used regression models like Linear Regression, XGBoost, MLP Regressor, and Random Forest to predict DC power directly. While these models showed good accuracy, they were not ideal for time-series forecasting. To eliminate this limitation, we subsequently employed deep learning models LSTM, 1D CNN, and GRU to predict key features like AC frequency, AC voltage, DC link voltage, energy today, output current, total energy, output power, DC, pyranometer reading, temperature, and power factor based on date and time as input. These forecasted features were further used to predict the target variable, DC Power. The LSTM model most accurately predicted the intermediate features and the end DC power output among the deep learning models. Our LSTM network was trained on real data and achieved a Root Mean Squared Error (RMSE) of 0.039, a Mean Absolute Error (MAE) of 0.020, and an R^2 score of 0.909, indicating high prediction accuracy.

Keywords: Solar Energy Prediction, LSTM, Time Series Forecasting, DL , Renewable Energy Management.

1. Introduction

Solar power has become a key component of sustainable energy transition policies, providing a scalable and renewable means for meeting the rising power needs of the world. As solar irradiance is inherently intermittent in nature in terms of patterns of weather and geography, reliable DC power output forecasting is required for increasing grid stability, energy dispatch, and load management. [1]. Traditional machine learning algorithms such as Linear Regression, Random Forest, XGBoost, and MLP Regressor have demonstrated effectiveness in static predictive tasks but lag in abstraction of temporal dependencies critical for time-series prediction[2]. For instance, although Linear Regression minimizes relations through linear approximations and aggregation methods such as XGBoost refine feature-based estimation, such methods cannot natively learn in series or long-patterns[2][3]. This limitation becomes particularly relevant when solar power systems must be customized for varying environmental and operating conditions. The temporal modelling challenge has triggered interest

surrounding deep learning architectures, specifically Long Short-Term Memory (LSTM) networks, which excel in detecting long-range dependencies in sequence data[4].

LSTM networks, introduced by Hochreiter and Schmid Huber, are best suited to handle the vanishing gradient problem and retain information over long sequences and, hence, are very effective for solar power time-series forecasting[4]. LSTM-based models have been proven in recent research to be superior to traditional machine learning models in solar energy metric prediction, with higher accuracy and better cyclical and seasonal trend management[5][6]. Furthermore, hybrid models that combine LSTM with convolutional layers or attention mechanisms could capture spatiotemporal features from photovoltaic data, albeit such models remain under-researched in DC power forecasting [6]. Despite all these advancements, functional barriers exist to translating predictive models into real-time decision-support systems. Published work often prioritizes computational accuracy over usability, limiting integration with user-friendly end-



user interfaces for solar farm managers and grid operators. The gap signifies a need for systems that allow proactive energy management based on high predictive performance and information dissemination via simple platforms. This work provides a two-level prediction model where LSTM feature prediction is coupled. During the first phase, LSTM networks predict intermediate features such as AC voltage, output current, and temperature using date and time-based temporal inputs. The predicted features serve as input for the second LSTM-based predictive model for DC power.

2. Literature Survey

Solar DC power generation forecasting has evolved significantly in the past decade with the momentum provided by the necessity to create value for solar power systems within the backdrop of rising global energy demand. Machine learning and statistical frameworks formed the basis of methods in the early years but failed to deliver a response to the intrinsic nature of solar data. Linear Regression (LR), for example, was readily applied since it is simple to establish linear relationships between environmental parameters (e.g., irradiance, temperature) and power output [7]. However, the fact that it could not handle nonlinear interactions and time correlations restricted its use to static solar systems [8]. Conversely, Random Forest (RF) and XGBoost both enhanced precision through ensemble learning and gradient boosting, but their fixed feature-based models were unable to leverage sequential patterns that are prevalent in time-series forecasting [[9], [10]].

A revolution accompanied deep learning, as deep learning architecture such as Multilayer Perceptrons (MLPs) and Long Short-Term Memory (LSTM) networks bypassed this limitation. MLPs could approximate nonlinear relationships but had difficulty with longer-distance temporal dependencies since they're feedforward [11]. In comparison, LSTM networks, which belong to the recurrent neural networks (RNNs) category, were a better choice for sequence data. Relying on memory cells and gate mechanisms, LSTMs could detect the long-range dependency of solar irradiance as well as power output variability [[12], [13]]. Husein & Chung (2019) [12] and Veerasamy et al. (2021)[13] conducted experiments that proved the efficacy of LSTMs with high day-ahead forecasting and fault detection in PV systems. Recent hybrid models, such as CNN-LSTM models, improved further by tapping both spatial feature extraction (using CNNs) and temporal modelling (using LSTMs) [14,15,16]. Despite such developments, real-time deployment and scalability gaps exist. Most research is based on algorithmic enhancement without integration in real-world applications into energy management systems. For

instance, systematic reviews by Al-Dahidi et al. (2024)[9] and Cantillo-Luna et al. (2023)[10] highlight the gap in bridging theoretical models and operational tools through frameworks [14]. In addition, existing research tends to overlook multivariate feature engineering essential to solar forecasting. Although past research tended to be irradiance- or temperature-centred [11], using this work of research, the feature set commonly shared across it includes DC link voltage, output current, and pyranometer input to attempt to improve model robustness [15]. The research strategy resonates with present times' findings to support all-inclusive monitoring of environmental and electrical parameters in a desire to predict better [16].

The shift from statistical models to deep learning is a journey towards managing solar data's complexity and time-dependent nature. Real-world deployment and feature engineering remain difficult. This work pushes solar forecasting toward real-time application scalability by integrating LSTM's temporal representation with Flutter's accessibility, closing critical gaps between academic and industrial applications.

2.1. Problem Statement

While previous research activity has applied numerous machine learning model-based systems in solar energy system DC power prediction, they will not perform satisfactorily, where it involves manipulating time-series data and recalling temporal relationships for lengthy periods. Such limitations undermine their predictability even more so when dynamic systems are exposed to real-time weather patterns. To address these issues, our research proposes an LSTM-based prediction model to address temporal patterns in DC power generation.

3. Related Work

3.1. Linear Regression

Linear Regression is a simple regression model algorithm that identifies the relationship between independent and dependent variables. The model implies that there is a linear relationship between the input features and output and that it can be expressed mathematically as:

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \dots + \beta_n X_n + \epsilon \quad (\text{Equation 1})$$

Y is the predicted DC power, X_i is the input features, β_i is the regression coefficients, and ϵ is the error term.

3.2. Random Forest

Random Forest is an ensemble learning algorithm that creates several Decision Trees and then takes their output. Every tree is trained on a random data subset (bootstrap

sampling), and the predictions are averaged.

$$\hat{Y} = \frac{1}{M} \sum_{j=1}^M \hat{Y}_j \quad (\text{Equation 2})$$

Where M is the size of the forest. Random Forest generalizes better by minimizing overfitting by averaging numerous trees.

3.3. XGBoost (Extreme Gradient Boosting)

XGBoost is a strong boosting algorithm that incrementally builds trees, refining the errors of the previous tree. It tries to reduce the following objective function.

$$\text{Obj} = \sum_{i=1}^n l(y_i, \hat{y}_i) + \sum_{k=1}^K \Omega(T_k) \quad (\text{Equation 3})$$

3.4. Multilayer Perceptron (MLP) Regressor

MLP is a feedforward neural network consisting of numerous layers of neurons. Each layer computes a weighted sum and an activation function:

$$h_i = \sigma(W_i X + b_i) \quad (\text{Equation 4})$$

We are the weights, b_i is the bias, and σ is the activation function (ReLU or Sigmoid).

3.5. Long Short-Term Memory (LSTM) Networks

LSTM networks are Recurrent Neural Networks (RNN) that excel at time-series forecasting by identifying long-term dependencies. Every LSTM cell is comprised of the following:

Forget Gate: Controls which past information to discard:

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f) \quad (\text{Equation 5})$$

Input Gate: Updates cell state with new information:

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \quad (\text{Equation 6})$$

$$\tilde{C}_t = \tanh(W_c \cdot [h_{t-1}, x_t] + b_c) \quad (\text{Equation 7})$$

Cell State Update: Combines old and new information:

$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t \quad (\text{Equation 8})$$

Output Gate: Computes the final output:

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) \quad (\text{Equation 9})$$

$$h_t = o_t \cdot \tanh(C_t) \quad (\text{Equation 10})$$

W are weight matrices, b are the biases, and σ is the sigmoid function.

3.6. Gated Recurrent Unit (GRU)

GRU is a Recurrent Neural Network (RNN) type that overcomes vanishing gradients using gating units. Unlike the conventional RNNs, GRUs better preserve long-term dependencies with fewer parameters than LSTM and are computationally efficient.

The GRU consists of two primary gates: the update and reset gates.

Update Gate: $z_t = \sigma(W_z x_t + U_z h_{t-1} + b_z)$ (Equation 11)

Reset Gate: $r_t = \sigma(W_r x_t + U_r h_{t-1} + b_r)$ (Equation 12)

Candidate Hidden State:

$$\tilde{h}_t = \tanh(W_h x_t + U_h(r_t \odot h_{t-1}) + b_h) \quad (\text{Equation 13})$$

Final Hidden State:

$$h_t = (1 - z_t) \odot h_{t-1} + z_t \odot \tilde{h}_t \quad (\text{Equation 14})$$

Here,

$w_z x_t$: Input vector at time step t

h_t : Hidden state at time step t

h_{t-1} : Previous hidden state

σ : Sigmoid activation function

$\tanh$: Hyperbolic tangent activation function

W, U, b: Learnable weight matrices and bias

vectors

3.7. 1D Convolutional Neural Network (1D CNN)

1D CNN extracts spatial features from sequences such as time-series data. It uses filters along one spatial dimension (typically time), and thus, it is best for modelling local dependencies.

$$y_i = \sum_{j=0}^{k-1} \sum_{c=0}^{C-1} w_{j,c} \cdot x_{i,j,c} + b \quad (\text{Equation 15})$$

Here:

y_i : Output feature map at position i

$w_{j,c}$: Filter weight at position j and channel c

$x_{i,j,c}$: Input value at corresponding position and channel

S: stride

3.8. Root Mean Squared Error (RMSE)

RMSE measures the standard deviation of prediction errors and can be defined as:

$$\text{RMSE} = \sqrt{\frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2} \quad (\text{Equation 16})$$

y_i are the actual values, $\hat{y}_i$ are the predicted values, and n is the number of observations. A lower RMSE represents higher model accuracy.

3.9. Mean Absolute Error (MAE)

MAE measures the mean absolute difference between actual and predicted values. A lower MAE value suggests

that the model has fewer absolute errors and better predictive performance.

$$MAE = \frac{1}{n} \sum_{i=1}^n |y_i - \hat{y}_i| \quad (\text{Equation 17})$$

3.10. R-squared score (R² Score)

R² estimates the extent to which the independent variables explain the variance of the target variable and is determined using:

$$R^2 = 1 - \frac{\sum (y_i - \bar{y})^2}{\sum (y_i - \bar{y})^2} \quad (\text{Equation 18})$$

Where $\bar{y}$ represents the mean of the actual values, an R² closer to 1 indicates a better fit of the model, i.e., the model explains most of the variance in the target variable.

4. Research Methodology

The methodology involves machine learning (ML) algorithms for feature prediction

4.1. Machine Learning Pipeline

Data Collection: The dataset used in this study contains real-time data obtained from solar power stations with several properties ranging from AC voltage to AC frequency and from pyranometer to energy today, with the extra advantage of including timestamps (Date and Time) that track the time values of such properties.

Data Preprocessing : Preprocessing is performed through the following steps:

Missing value handling by imputation techniques. Scaling the numerical features using Min-Max normalization so that all the points fall into the same range. Transforming Date and Time into numeric forms appropriate for temporal analysis. Zero values are kept where they matter, like idle or minimal power generation.

Feature Engineering: Temporal features like day of hour, day of week, and month were extracted from DATE and TIME fields to capture trends over time. Correlation analysis was also utilized to obtain top-influencing features to serve as predictors in the second-stage model.

Model Selection & Training: Using input features, some machine learning models were employed to predict the DC Power. The models are: Linear Regression (LR), Multilayer Perceptron (MLP), Random Forest (RF), XGBoost (XGB)

While these models provided a sound output for direct prediction, they were insufficient to capture the temporal dependencies in the data, and hence, further research was carried out on using time-series forecasting models.

Experiments were conducted with LSTM, GRU, and 1D CNN models to achieve accurate feature predictions. These deep learning models are optimally suited for time-series data since they can readily identify long dependencies. On testing, LSTM was the best and was used subsequently.

4.2. DC Power Prediction Using LSTM

Once the input features are forecasted, the forecasted values train an LSTM model to predict the DC power output. LSTM's ability to learn sequential dependencies is instrumental in delivering accurate power predictions in solar systems.

4.3. Model Evaluation

To assess model performance, the following were employed as metrics:

Root Mean Squared Error (RMSE): RMSE measures the standard deviation of prediction errors.

Mean Absolute Error (MAE): MAE measures the mean absolute difference between actual and predicted values.

R² Score (Coefficient of Determination): R² estimates the extent to which the independent variables explain the variance of the target variable.

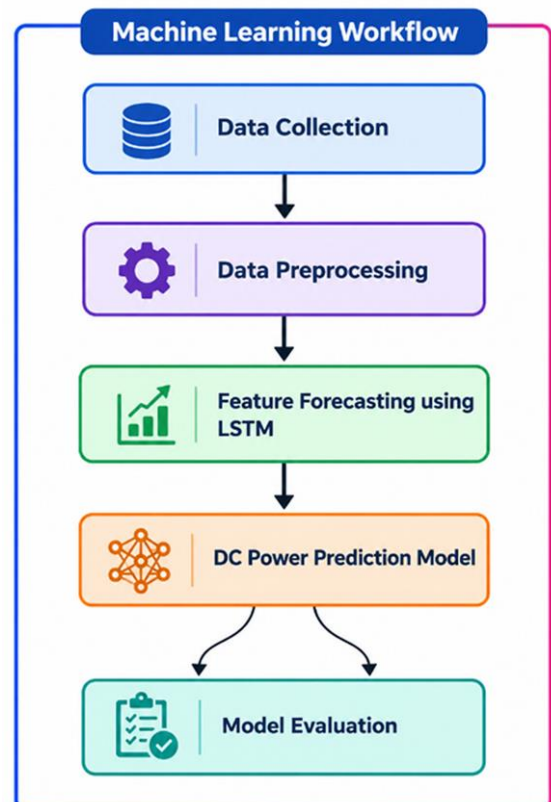


Figure. 1 Methodology

Therefore, LSTM performed better than all the others and was the final deployment choice.

5. Experimental Implementation

5.1. Dataset Description

Source:

The DC Power data has been obtained from a solar power plant. The data has been obtained to model DC power generation as a function of various environmental and system variables such as temperature, irradiance, and energy consumption measurements.

Attributes:

The most critical characteristics of the data are:

AC Frequency: Alternating current (AC) supply frequency.

AC Voltage: Alternating current voltage.

DC Link Voltage: Voltage on the DC link, connecting the solar array and the inverter.

Energy Today: Energy generated today in kilowatt-hours (kWh).

Output Current: Inverter current output.

Total Energy: Total energy output from the solar plant.

Output Power: Output power in watts.

DC Current: DC-side current of the system.

Pyranometer: Solar irradiance measurements.

Temperature: Ambient temperature surrounding the solar panel.

Power Factor: A measure of the efficiency in power usage.

Period:

The information includes data from 01-01-2019 to 31-12-2019. The time intervals are in the format of HH:MM:SS.

Size:

The data has 101951 records, with a sampling period of 5 minutes. This means the data points are recorded regularly, allowing for temporal analysis.

Table. 1 Attributes/Parameters and its ranges

S.NO	PARAMETERS	RANGE OF VALUES
1	DATE	01-01-2019 to 31-12-2019
2	TIME	00:00:00 to 23:55:00
3	AC FREQUENCY	-58.210 to 50.290 (Hz)
4	AC VOLTAGE	0.000 to 460.000(V)
5	DCLINK VOLTAGE	0.000 to 823.000(V)
6	ENERGY TODAY	0.000 to 453.000(kWh)
7	OUTPUT CURRENT	0.000 to 121.200(A)
8	TOTAL ENERGY	0.000 to 20033.000(kWh)
9	OUTPUT POWER	0.000 to 82.900(W)
10	DC CURRENT	-75.00 to 143.343(A)
11	PYRANOMETER	-0.694 to 1044.347(W/m ²)
12	TEMPERATURE	14.497 to 44.794(°C)
13	DC POWER	-60533.750 to 90103.914(W)
14	POWER FACTOR	0.0000 to 0.983

5.2. Preprocessing

Normalization/Scaling: Data was scaled using the Min-Max Scaler to aid model convergence. All the features were rescaled from [0, 1] to avoid a specific feature dominating model learning since the data had different scales.

Data Splitting: The data were split into a training set and a test set using the 80-20 split so that 80% of data were utilized in model training, and the remaining 20% leftover was reserved as test data. In this case, the splitting was temporal since a time series predictive task is one where the proper order of events is essential.

5.3. Experimental Setup

Libraries Used:

TensorFlow: For training and developing the LSTM model.

Scikit-learn: To preprocess the data, divide the data, and evaluate the model.

Matplotlib, Seaborn: To visualize and for plotting results.

Pandas, NumPy: To manipulate and work with the data.

5.4. Evaluation Metrics

MAE (Mean Absolute Error): An average measure of the size of the prediction errors in absolute but not direction. It is a good indicator of how well the model performs in reality.

RMSE (Root Mean Squared Error): RMSE approximates the square root of the mean of the squared errors and assigns more importance to more significant errors than MAE. It helps evaluate the model's goodness to predict the magnitude of DC power.

R^2 Score: The R-squared score indicates the variance in the target variable (DC Power) explained by the features. The higher the R^2 score, the better the model fits the data.

5.5. Model Training

Hyperparameter Tuning:

Number of Layers: Two 50-neuron LSTM layers are utilized in the model. The first LSTM layer returns sequences to the next LSTM layer, and the second LSTM layer makes the final prediction.

Neurons per Layer: Each LSTM layer contains 50 neurons.

Batch Size: The batch size was 32 when training.

Learning Rate: The learning rate was optimized to 0.0005 for faster convergence.

Optimizer: We used the Adam optimizer to minimize the loss function.

Cross-Validation: We did not use cross-validation as the data was time series. Instead, we performed a holdout validation with an 80-20 train-test split, using 80% of the data for training and 20% for testing.

6. Results & Analysis Discussion

6.1. Performance Metrics

MAE is the mean absolute difference between predicted and actual values, providing a straightforward measure of prediction error. RMSE is skewed towards more significant

errors by squaring differences first and thus is outlier sensitive. R^2 Score measures the fit of estimated values to actual results, where a higher rating is near the better fit. Together, these two measurements form a composite measure of the model's performance regarding precision and prediction variance.

6.2. Comparative Analysis of Regression Models

The comparative evaluation was conducted between four regression models—Linear Regression, Random Forest, XGBoost, and Multilayer Perceptron (MLP) Regressor—using MAE, RMSE, and R^2 as evaluating metrics. Results are shown in Table II.

Among the models, the MLP Regressor had the lowest RMSE (136.60), indicating that it can reduce prediction errors to the lowest extent. Random Forest and XGBoost had the highest R^2 values (0.9999), indicating their ability to identify complex patterns in data, with Random Forest having the lowest MAE value (36.99) as well. Although Linear Regression had a high R^2 value (0.9997), high error values indicate non-proficiency in handling nonlinear behaviour in the data.

As DC power generation involves being time-dependent in form by nature (e.g., solar trends, weather patterns), applying non-time series models alone might not be capable of capturing these trends [32]. Hence, we utilized models in the following analysis phase to enhance capability and leverage sequent leverages.

Table 2: Comparison of performance of Regression Models

Model	MAE	RMSE	R^2 Score
Linear Regression	199.4198	387.80	0.9997
Random Forest	36.9876	243.44	0.9999
XGBoost	57.6772	200.0539	0.9999
MLP Regressor	95.00	136.6034	0.9996

6.3. Comparative Analysis of Forecasting Models

To overcome the limitation of conventional non-time series models in terms of temporal dependencies, forecasting models were utilized to forecast input features of previous dates and times. Subsequently, the final DC power output was estimated from predicted features. The performance evaluation of three forecasting models based on deep learning, namely LSTM, GRU, and 1D-CNN, are compared using the RMSE, MAE, and R^2 metrics, depicted in Table II.

Among the models of predictions, the LSTM model performed the best with the lowest RMSE of 0.015, MAE of 0.037 and the highest R^2 of 0.918, thus having greater prediction accuracy and generalization capacity. This

reflects the capacity of the LSTM to capture temporal dependencies and sequential patterns from the input. On the other hand, GRU obtained an R^2 of 0.897 but with much more significant error values, indicating a comparatively less stable performance over time. The lowest R^2 value among the 1D-CNN model (0.842) indicates high error values, indicating failure to capture the sequential nature of the data well.

These findings emphasize the significance that prediction based on LSTM is most suitable for DC power prediction if time-dependent features are utilized, with higher accuracy and legitimacy compared to GRU and CNN models.

6.4. LSTM Model Architecture

A long Short-Term Memory (LSTM) network is a Recurrent Neural Network (RNN) that can learn long-term

dependencies and is particularly well-suited for time series forecasting tasks such as DC power forecasting.

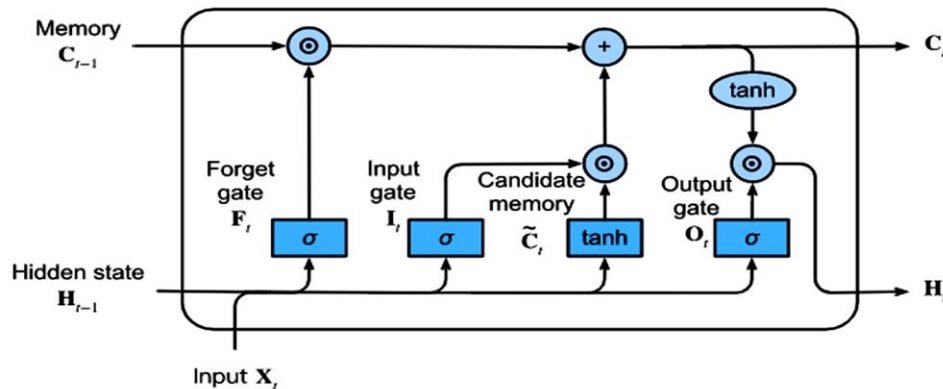


Figure 2: LSTM architecture

As can be observed in Figure 2, there are three gates in the LSTM unit: A) Forget Gate (F_t): Regulates what memory cell information must be forgotten. B) Input Gate (I_t): Determines what new information to store in the cell state. C) Output Gate (O_t): Determines what portion of the cell state to output as the hidden state. The cell state (C_t) passes the memory so that the network can hold significant

time-dependent features, and the hidden state (H_t) is utilized for prediction. Sigmoid (σ) and tanh are used as nonlinear functions in the gates to control the flow of information. The architecture allows LSTM to effectively address the temporal character of solar energy data and make more accurate predictions of future DC power output.

Table 3: Comparison of performance of deep learning models

Model	RMSE	MAE	R ² score
LSTM	0.015	0.037	0.918
GRU	5535.259	2543.691	0.897
1D-CNN	3928.321	7651.344	0.842

6.5. Scatter Plot Analysis

Fig. 3 is the actual vs. predicted scatter plot of DC power values collected using the LSTM model. The dashed diagonal line signifies the best-case scenario where the expected values fit perfectly with the actual values. The

points lie almost entirely on or very near this line, exhibiting good correlation and robust prediction behaviour. There is little scattering at low powers, possibly resulting from noise or variations in environmental conditions during such periods. The model is of excellent accuracy, particularly with mid-to-high DC power.

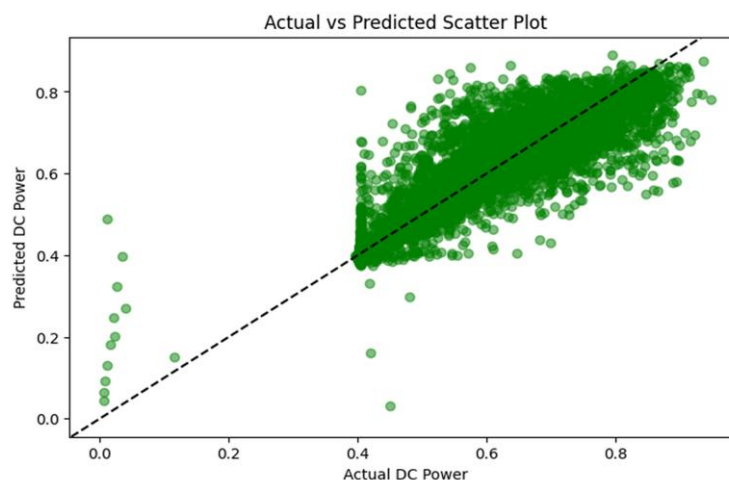


Figure 3: Scatter Plot of Actual vs Predicted values of LSTM model

6.6. Training and Validation Loss Curve

Fig. 4 shows the training and validation loss of the LSTM model over epochs as Mean Squared Error (MSE). The plot for training loss decreases slowly continuously and sharply initially before declining slowly, reflecting good learning. The validation loss decreases with slight

oscillation after the second epoch. The low gap between training and validation curves reflects enough generalization with little overfitting. These results verify the stability of the model during training and its performance on new data.

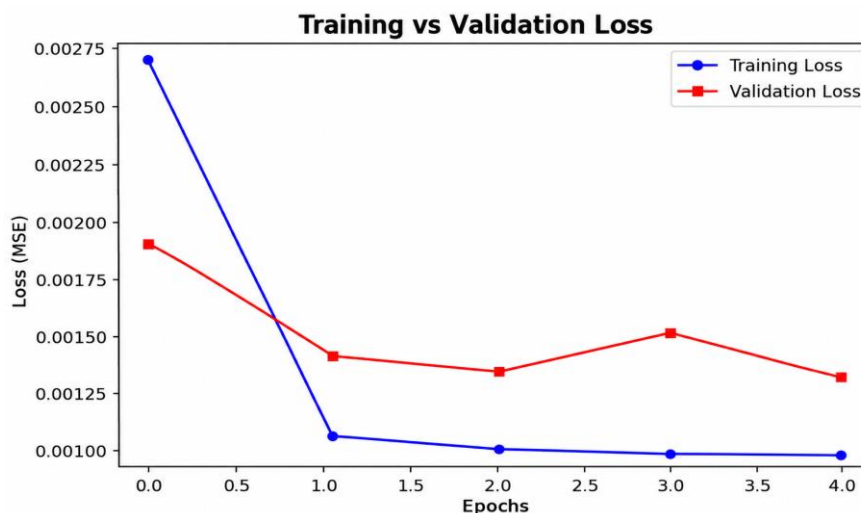


Figure. 4 Training Vs Validation Loss Curve of LSTM model

6.7. Error Distribution Analysis

Fig. 5 shows the residual plot of the LSTM model, i.e., the difference between actual and predicted DC power values. Most errors are clustered around zero, meaning the

model's predictions are accurate and unbiased. The plot appears tight and well-balanced, with hardly any outliers, which guarantees the stability and reliability of the model's forecasting power. This type of residual profile is typical of a well-generalized and good predictive model.

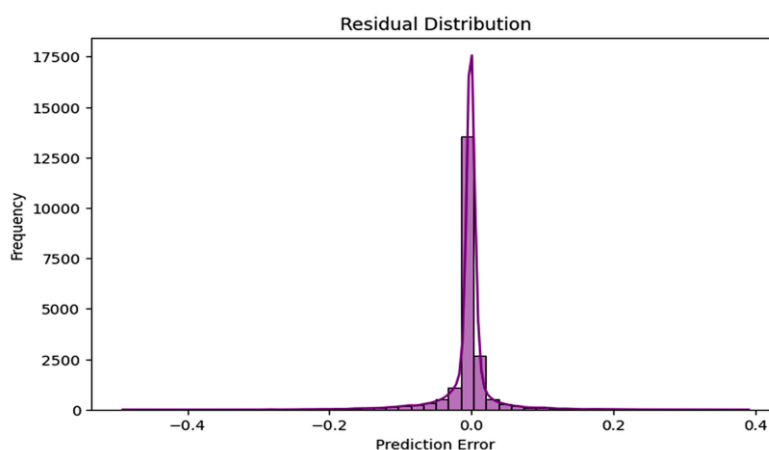


Figure. 5 Residual Distribution

7. Conclusion and Future Scope

This paper brings forward an aggressive prediction model for DC power forecasting in solar energy systems from a broad range of models, from simple machine learning models, including Linear Regression, Random Forest, and XGBoost, to advanced deep learning networks, including Gated Recurrent Units (GRU), 1D Convolutional Neural

Networks (1D CNN), and Long Short-Term Memory (LSTM) networks. The GRU and LSTM models performed better in learning temporal dependencies, while 1D CNN efficiently extracted local patterns from the input features to improve overall prediction accuracy. The last LSTM model, chosen based on its performance in reducing error metrics (RMSE, MAE, and R^2 score)

The research contributes significantly towards real-time solar power monitoring, predictive maintenance, and grid optimization by combining comparative model analysis. Future work includes projecting the model's scalability to big data, employing edge computing for offline execution, and applying the framework to hybrid energy systems to generalize renewable energy forecasting.

How To Cite:

Chigilipalli Bharat Kumar , " Integrating LSTM and Flutter For Real-Time Solar DC Power Prediction " , International Journal of Research and Development in Engineering Sciences (IJRDES), Volume 8, Issue 4 , pp 8 - 17 , October 2026. CrossRef DOI: <https://doi.org/10.63328/ijrdes-v8ri5p2>

References

- [1]. Hyndman, R. J., & Athanasopoulos, G. (2018). *Forecasting: Principles and Practice*. OTexts.
- [2]. Breiman, L. (2001). Random Forests. *Machine Learning*, 45(1), 5–32.
- [3]. Friedman, J. H. (2001). Greedy Function Approximation: A Gradient Boosting Machine. *Annals of Statistics*, 29(5), 1189–1232.
- [4]. Hochreiter, S., & Schmidhuber, J. (1997). Long Short-Term Memory. *Neural Computation*, 9(8), 1735–1780.
- [5]. Liang, X., et al. (2020). Solar Power Forecasting Based on LSTM Neural Network. *Energy Reports*, 6, 272–286.
- [6]. Wang, Y., et al. (2022). Hybrid Deep Learning Models for Solar Power Forecasting: A Review. *Renewable and Sustainable Energy Reviews*, 153, 111789.
- [7]. Al-Janabi, S., & Al-Janabi, Z. (2023). Development of deep learning method for predicting DC power based on renewable solar energy and multi-parameters function. *Neural Computing and Applications*, 35(21), 15273-15294.
- [8]. Akhter, M. N. (2021). *Analysis of Photovoltaic Panels Performance and Power Output Forecasting Based on Optimized Deep Learning Technique* (Doctoral dissertation, University of Malaya (Malaysia)).
- [9]. Alhosani, A. (2023). Development and Application of Forecasting Machine Learning Approach for Renewable Energy-Related Products' Prices (Doctoral dissertation, Khalifa University of Science).
- [10]. Al-Dahidi, S., Madhiarasan, M., Al-Ghussain, L., Abubaker, A. M., Ahmad, A. D., Alrbai, M., ... & Zio, E. (2024). Forecasting solar photovoltaic power production: A comprehensive review and innovative data-driven modelling framework. *Energies*, 17(16), 4145.
- [11]. Cantillo-Luna, S., Moreno-Chuquen, R., Celeita, D., & Anders, G. (2023). Deep and machine learning models to forecast photovoltaic power generation. *Energies*, 16(10), 4097.
- [12]. Sudharshan, K., Naveen, C., Vishnuram, P., Krishna Rao Kasagani, D. V. S., & Nastasi, B. (2022). A systematic review on the impact of different irradiance forecasting techniques for solar energy prediction. *Energies*, 15(17), 6267.
- [13]. Husein, M., & Chung, I. Y. (2019). Day-ahead solar irradiance forecasting for microgrids using a long short-term memory recurrent neural network: A deep learning approach. *Energies*, 12(10), 1856.
- [14]. Veerasamy, V., Wahab, N. I. A., Othman, M. L., Padmanaban, S., Sekar, K., Ramachandran, R., ... & Islam, M. Z. (2021). LSTM recurrent neural network classifier for high impedance fault detection in solar PV integrated power system. *IEEE Access*, 9, 32672-32687.
- [15]. Ledmaoui, Y., El Fahli, A., El Maghraoui, A., Hamdouchi, A., El Aroussi, M., Saadane, R., & Chebak, A. (2024). Enhancing solar power efficiency: Smart metering and ANN-based production forecasting. *Computers*, 13(9), 235.
- [16]. Liu, Y. T., & Tang, K. T. (2020). A minimum distance inlier probability (MDIP) feature selection method to improve gas classification for electronic nose systems. *IEEE Access*, 8, 133928-133935.
- [17]. Qaiyum, S., Margala, M., Kshirsagar, P. R., Chakrabarti, P., & Irshad, K. (2023). Energy performance analysis of photovoltaic integrated with microgrid data analysis using deep learning feature selection and classification techniques. *Sustainability*, 15(14), 11081.
- [18]. Kumar, A., Kashyap, Y., & Kosmopoulos, P. (2022). Enhancing Solar Energy Forecast Using Multi-Column Convolutional Neural Network and Multipoint Time Series Approach. *Remote Sens.* 2023, 15, 107.
- [19]. Abdullah, B. U. D., Khanday, S. A., Islam, N. U., Lata, S., Fatima, H., & Nengroo, S. H. (2024). Comparative Analysis Using Multiple Regression Models for Forecasting Photovoltaic Power Generation. *Energies*, 17(7), 1564.
- [20]. Elizabeth Michael, N., Mishra, M., Hasan, S., & Al-Durra, A. (2022). A short-term solar power prediction model uses a multi-step CNN stacked LSTM technique. *Energies*, 15(6), 2150.
- [21]. Wu, Y. K., Huang, C. L., Phan, Q. T., & Li, Y. Y. (2022). I completed a review of various solar power forecasting techniques, considering different viewpoints. *Energies*, 15(9), 3320.
- [22]. Khandakar, A., EH Chowdhury, M., Khoda Kazi, M., Benhmed, K., Touati, F., Al-Hitmi, M., & Jr SP Gonzales, A. (2019). Machine learning-based photovoltaics (PV) power prediction using different environmental parameters of Qatar. *Energies*, 12(14), 2782.
- [23]. Joshua, S. R., Yeon, A. N., Park, S., & Kwon, K. (2024). A Hybrid Machine

Learning Approach: Analyzing Energy Potential and Designing Solar Fault Detection for an AIoT-Based Solar-Hydrogen System in a University Setting. *Applied Sciences*, 14(18), 8573.

- [24]. Hassan, I., Alhamrouni, I., Younes, Z., Azhan, N. H., Mekhilef, S., Seyedmahmoudian, M., & Stojcevski, A. (2024). Explainable deep learning model for grid-connected photovoltaic system performance assessment for improving system reliability. *IEEE Access*.
- [25]. Gaviria Fierro, J. F. (2021). Machine learning in photovoltaic systems.
- [26]. Rauf, M. Q. (2020). Photovoltaic Power Forecast in an Energy Management System for a Hybrid Microgrid (Doctoral dissertation, CAPITAL UNIVERSITY).
- [27]. Neural Network and Multipoint Time Series Approach. *Remote Sens.* 2023, 15, 107.
- [28]. Dairi, A., Harrou, F., Sun, Y., & Khadraoui, S. (2020). Short-term photovoltaic solar power production forecasting using variational auto-encoder driven deep learning approach. *Applied Sciences*, 10(23), 8400.
- [29]. Balouji, E., Gu, I. Y., Bollen, M. H., Bagheri, A., & Nazari, M. (2018, May). An LSTM-based deep learning method with application to voltage dip classification. In *2018 18th International Conference on Harmonics and Quality of Power (ICHQP)* (pp. 1-5). IEEE.
- [30]. Tavakolizadeh, N., & Bagheri, M. (2022). Multi-attribute selection for salt dome detection based on SVM and MLP machine learning techniques. *Natural Resources Research*, 31(1), 353-370.
- [31]. Alazab, M., Khan, S., Krishnan, S. S. R., Pham, Q. V., Reddy, M. P. K., & Gadekallu, T. R. (2020). A multidirectional LSTM model for predicting the stability of a smart grid. *Ieee Access*, 8, 85454–85463.
- [32]. Lei, T. M., Ng, S. C., & Siu, S. W. (2023). Application of ANN, XGBoost, and other ML methods to forecast air quality in Macau. *Sustainability*, 15(6), 5341.

Declaration

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

Author Contribution: All authors wrote the main manuscript text and also consent to the submission.

Ethical approval: Not applicable.

Consent to Participate: All authors consent to participate.

Funding: Not applicable, and No funding was received

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Personal Statement: We declare with our best of knowledge that this research work is purely Original Work and No third party material used in this article drafting. If any such kind material found in further online publication, we are responsible only for any judicial and copyright issues.

Generative AI Statement: The authors declare that generative AI was not used in the preparation or creation of this manuscript. The alternative text (alt text) accompanying the figures in this article was generated by Frontiers with the assistance of artificial intelligence. Reasonable efforts have been made to ensure the accuracy and appropriateness of the generated alt text, including review by the authors wherever possible. If you identify any inaccuracies or issues, please contact the editorial team.

Publisher's Note: The statements, opinions, and claims presented in this article are exclusively those of the authors and do not necessarily reflect the views of their affiliated institutions, the publisher, editors, or reviewers. Any products discussed or evaluated, as well as any claims made by their manufacturers, are not warranted, approved, or endorsed by the publisher.

Acknowledgements

We thank everyone who inspired our work