JOURNAL OF RENEWABLE ENERGY AND SMART GRID TECHNOLOGY

Vol. 21, No. 1, January-June 2026

Sliding Window-Based Deep Learning Approach for Solar Power Forecasting in Malaysia Utility-Scale PV Systems

Noor Hasliza Abdul Rahman[1,4*], Shahril Irwan Sulaiman[2,3], Muhammad Asraf Hairuddin[1,5], Mohamad Zhafran Hussin[1,6], Ezril Hisham Mat Saat[1]

[1]Faculty of Electrical Engineering, Universiti Teknologi MARA, Johor Branch, Pasir Gudang Campus, 81750 Masai, Malaysia
[2]Faculty of Electrical Engineering, Universiti Teknologi MARA, 40450 Shah Alam, Malaysia
[3]Green Energy Research Centre, Universiti Teknologi MARA, 40450 Shah Alam, Malaysia
[4]SPECTRA Research Group, Faculty of Applied Sciences, Universiti Teknologi MARA, 40450 Shah Alam, Malaysia
[5]Institute for Big Data Analytics and Artificial Intelligence (IBDAAI), Universiti Teknologi MARA, 40450 Shah Alam, Malaysia
[6]Solar Research Institute (SRI), Universiti Teknologi MARA, 40450 Shah Alam, Malaysia
***Corresponding author's email**: noorhasliza@uitm.edu.my

Article info:

Received: 10 July 2025
Revised: 15 October 2025
Accepted: 21 October 2025
DOI:
10.69650/rast.2026.263014

Keywords:

Sliding Window
Deep Learning
Utility-Scale Photovoltaic
Large-Scale Solar
Solar Power Forecasting

ABSTRACT

Accurate forecasting of solar power in utility-scale photovoltaic (USPV) systems is critical for grid stability but remains challenging due to meteorological variability and the large spatial scale of these systems. However, the choice of sliding window size in time-series forecasting remains underexplored. This study introduces a deep learning-based forecasting framework that systematically evaluates the impact of sliding window size on forecasting accuracy using multivariate time-series data. The data collected from a 25 MWac USPV system in Malaysia between August 2022 and April 2023, comprises 5-minute interval measurements of solar irradiance, module temperature and solar power output. Multiple deep learning (DL) models, namely LSTM, CNN and GRU across window sizes ranging from 12 to 288 steps and forecasting horizons of 1 to 12 hours were investigated. Results show that a 144-step window consistently improves accuracy over conventional one-step input methods, with LSTM outperforming other models by achieving up to 23.1% RMSE reduction, 30.7% MAE reduction and a 8.6% increase in R^2 at 60 minutes forecasting horizon. This work emphasizes the importance of window size selection in optimizing forecasting accuracy for USPV systems and supporting renewable energy grid integration. By improving forecasting capabilities, this research is expected to provide critical insights to enhance renewable energy integration into the grid system.

1. Introduction

Solar photovoltaic (PV) systems represent a fundamental component of the global renewable energy (RE) landscape, owing to their scalability, low operating cost and continuous efficiency enhancements [1]. The increasing global commitment to reduce greenhouse gas emissions and reliance on fossil fuels has further accelerated the deployment of PV systems, positioning them as one of the fastest growing RE technologies worldwide.

In Malaysia, the rapid deployment of USPV systems is driven by government initiatives such as the Large Scale Solar (LSS) scheme and other renewable energy (RE) program aimed at increasing RE penetration in the national grid [2] . As of 2024, a total capacity of 2,445.372 MW has been awarded nationwide through successive LSS programs, from LSS1 to LSS5, thereby contributing significantly to the country's RE portfolio and supporting the national agenda for sustainable energy development [3].

Despite this progress, the intermittent and weather-dependent nature of solar irradiance results in substantial uncertainty in solar power output, presenting significant challenges for grid stability and operational planning. As such, accurate forecasting of solar power is vital for effective grid scheduling, economic dispatch, and seamless integration of variable renewable energy into the power system.

Recently, machine learning (ML) and deep learning (DL) models have been increasingly employed in solar power forecasting due to their ability to capture complex nonlinear relationships between meteorological and historical generation data [4-5]. These models have demonstrated superior performance compared to traditional statistical and physical models, particularly under conditions of variability [6-10]. However, while much attention has been devoted to developing advanced model architectures, the role of temporal input design, particularly the sliding window size, remains underexplored. Sliding window techniques which transform time series data for supervised learning, can significantly affect the forecasting performance by influencing how well the models capture short and long-term dependencies.

This study examines the effect of varying sliding window configurations on the forecasting performance of DL models, utilizing a comprehensive multivariate dataset from a 25 MWac USPV system in Malaysia. Through systematic experimentation across different window sizes, DL architectures and forecast horizons, the study contributes valuable insights into the optimization of temporal input structures for enhanced solar power forecasting.

2. Literature Review

Several studies were conducted to forecast solar power using ML and DL models. For instance, a simplified Long Short-Term Memory (LSTM) model was applied for day-ahead solar power forecasting [7]. The model was trained on datasets of varying durations of 1, 3 and 6 months, collected from PV installations in Kaohsiung, Taiwan, and Chiang Rai, Thailand. The results revealed minimal variation in the Root Mean Square Error RMSE metric across different dataset sizes. Despite the model's superior forecasting performance, the study did not comprehensively address the influence of the effect of varying training window sizes which a critical factor that may affect forecasting accuracy and model robustness [10].

Furthermore, an LSTM model has been employed for one-hour-ahead energy forecasting at a 1.15 MW solar power plant in Turkey [8]. This research utilizes an LSTM neural network and evaluates its performance against alternative data-driven methods, including adaptive neuro-fuzzy inference systems (ANFIS) with fuzzy c-means (FCM) and ANFIS with grid partitioning (GP). The results indicated that the LSTM model achieved superior forecasting accuracy, yielding an RMSE of 60.66 kWh, MAE of 30.47 kWh, and an R^2 value of 0.9777. Consistent with prior studies, this research exclusively utilized historical solar power data without incorporating additional weather variables and neglecting the optimization of input window size, which could play a critical role in improving both short and long-term forecasting accuracy.

Besides that, CNN architectures have been applied for forecasting hourly PV energy production in Romania [11] . The results indicate that CNN models significantly outperform LSTM models, with 77% of CNNs achieving a Coefficient of Determination (R^2) of 0.9 or higher compared to only 13% for LSTMs. The best-performing CNN model reached an R^2 of 0.9913 with a mean absolute error (MAE) of 9.74, while the top LSTM model achieved an R^2 of 0.9880 and an MAE of 12.57. Although these results highlight the superiority of CNNs in this context, the study did not address the determination of an optimal sliding window size, which remains a critical element in enhancing forecasting accuracy.

Beyond CNN, a large-scale solar study used hybrid GRU models to forecast solar power for a 75 MW system [12]. The research implemented macro-level and inverter-level forecasting methods to predict solar output 1–6 hours in advance. It also proposed a systematic heuristic method for optimizing hyperparameters, aimed at guiding unbiased model development. Results indicated a modest improvement in accuracy, especially with aggregated inverter-level forecasts. The macro-level model achieved Mean Absolute Percentage Error (MAPE) values between 1.42% and 8.13%, while inverter-level forecasts had slightly better MAPE values ranging from 1.27% to 8.29%. The study concluded that macro-level deep learning models effectively capture system behavior but suggested that these methods might be more suitable for large scale PV system below 10 MW.

A complementary fused method combining Gated Recurrent Unit (GRU) networks and Extreme Gradient Boosting (XGBoost) has been developed for long-term hourly solar energy forecasting [13]. The model was validated on the Tianchi UNiLAB competition dataset as well as GEFCom2014. The proposed approach achieved first place in the competition and outperformed state-of-the-art models, reducing Mean Square Error (MSE) and MAE by 28.3% and 17.4%, respectively, over a 150-step forecasting horizon. Despite these promising results, the study did not consider interpretability issues or optimize the input sliding window size.

Research to date shows that DL models excel at solving complex prediction problems across a wide range of application, particularly in time-series forecasting [14]. However, pre-processing is a crucial prerequisite for utilizing time series data. In the context of solar power forecasting, the unique characteristics of PV systems, such as the high nonlinearity and intermittent solar power generation driven by dynamic weather conditions, demand specialized pre-processing strategies. Among these, determining the optimal input window size plays a critical role in accurately capturing the temporal patterns of the data. Despite its importance, prior research has often overlooked a detailed exploration of methodologies for selecting the most effective window size.

Conventionally, time series forecasting employs fixed, one-step input window sizes across datasets [15]. Although widely used, it may not adequately account for the complex and varying temporal dependencies inherent in solar power data. Alternative approaches have been explored in [10] which applied a 12-step input window to evaluate the performance of a Convolution Neural Network - Long-Short Term Memory (CNN-LSTM) model in solar irradiance forecasting, while other research utilised a DNN model with a 7-step input window [16]. Besides that, a fixed-size sliding window mechanism is applied to prevent data leakage and decomposition-level problem during the decomposition process using LSTM [17]. A recent study by X.Cui et al. proposed an ensemble method with adaptive sliding window configurations [18]. These studies highlight the potential of varying window sizes, nevertheless, a framework for determining optimal window sizes remains lacking. An inappropriate selection input window size may fail to capture short-term or long-term temporal patterns, thereby diminishing forecasting accuracy and reliability. Hence, addressing these gaps forms the core motivation of the present work.

This paper presents the development of a DL-based approach, utilizing LSTM, CNN, and Gated Recurrent Unit (GRU) models. The study employs a multivariate real-world dataset from a 25 MWac USPV system in Malaysia. The proposed methodology incorporates multiple sliding window sizes during the pre-processing stage to effectively capture the intricate data characteristics spanning varying temporal scales. This approach aims to systematically determine the optimal input window size for each DL model. To evaluate the effectiveness of the proposed strategy, a comprehensive performance analysis is conducted. The study compares the optimal window size determined through this methodology with conventional one-step window methods, highlighting the advantages of optimal window size selection in enhancing forecasting accuracy and reliability.

The structure of this paper is as follows: Section 2 discusses the related research work that applies in USPV, Section 3 provides a detailed description of the USPV system site used in this study. Section 4 outlines the forecasting research methodology, including pre-processing, sliding window and DL model configurations. Section 5 presents and discusses the results of the experiments conducted. Finally, the paper concludes with Section 6, summarizing the key findings and suggesting directions for future work.

3. Utility-scale PV System Description

The case study in this research focuses on a real-world 25 MWac and 31.57 MWp USPV system located in the southern region of Malaysia, specifically in Pasir Gudang, Johor (latitude 1.5283°N, longitude 103.8759°E), as shown in Fig. 1. This PV plant is the second USPV system owned and operated by Universiti Teknologi MARA Malaysia, named UiTM Solar Park II.

The construction of the plant is started in February 2020 and achieved commercial operation status by December 2020. UiTM Solar Park II spans a vast area of 112 acres, facilitated by 94,260 PV modules and each rated at a maximum power output of 335 Wp. In order to ensure optimal energy conversion and system efficiency, the plant is supported by 160 inverters, each rated at 165 kW. To facilitate efficient operation and maintenance, the PV array is systematically segmented into operational zones. This zonal configuration enables localized monitoring, streamlined fault detection and efficient zonal-level maintenance strategies. The PV modules are installed on fixed ground-mounted structures tilted at 15° from the horizontal plane.

Fig. 1 Location of UiTM Solar Park II.

4. Methodology

The proposed methodology utilises DL approach incorporated with a multiple sliding window size to forecast solar power in USPV system as shown in Fig. 2. The framework consists of five stages: data acquisition, pre-processing, input sliding window construction, DL model implementation and performance evaluation. Initially ground-based measurement multivariate dataset such as solar irradiance, module temperature and historical AC solar power are collected from the USPV plant database. These datasets are then pre-processed through cleaning to prepare them for time-series analysis. A multiple sliding window size is applied next, generating overlapping data segments with varying window size (step lag) of 12, 24, 36, 60, 72, 144, and 288 to enable the models to learn both short-term and long-term temporal patterns. This pre-processed data from sliding window is then fed into three selected DL architectures specifically LSTM, CNN and GRU, each designed to perform solar power forecasting. The final stage involves determining the optimal sliding window sizes for each model by evaluating their forecasting performance using RMSE, MAE and R^2.

4.1 Data Acquisition

This study utilise a dataset comprising meteorological and historical solar power. Meteorological data, specifically solar irradiance and module temperature are collected using sensors installed at Meteorological Measuring Facilities (MMF) positioned strategically across the PV plant. Irradiance sensors are mounted at an angle aligned with the solar panel structures, enabling precise solar radiation measurements, while module temperature sensors are attached to the rear of the panels, providing precise measurements of the panels operating temperatures.

In addition to meteorological data, historical solar power data was retrieved from an inverter data logger. This data is transmitted in real time to the plant's monitoring center, where it is archived in a centralized database for extended analysis. In this study, data was collected at 5-minute intervals from August 1, 2022, to April 30, 2023, resulting in 87,552 data points for each variable. In contrast to earlier studies that restricted data collection to daytime intervals, 10 hours [19], 13 hours [20] , this study employs a comprehensive 24-hour dataset, encompassing both daytime and nighttime periods.

4.2 Data Pre-processing

Data preprocessing constitutes the second stage of the forecasting framework and plays an essential role in preparing a clean, consistent, and error-free dataset. The process initiates with data cleaning, including the imputation of missing values and the removal of outliers to mitigate the adverse effects of anomalies during training process. Missing data in critical parameters such as solar irradiance and module temperature typically result from technical malfunctions such as sensor failure could disrupt continuous data acquisition. To resolve this, data imputation is performed using an interpolation technique grounded in the Linear Correlation Approach (LCA).

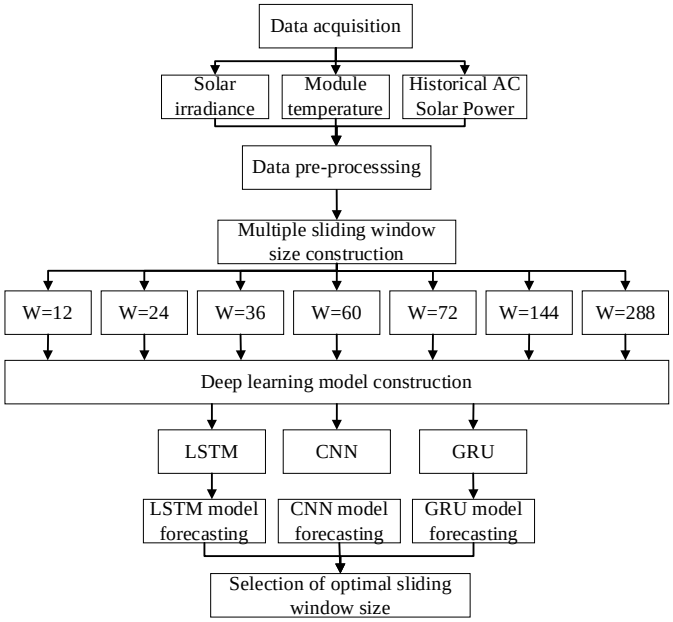

Fig. 2 Framework of the proposed DL approach utilizing a multiple window size for solar power forecasting.

The LCA leverages the strong empirical relationship between PV solar power (Pac) and the parameters with missing values, namely solar irradiance (G) and module temperature (T). Unlike traditional interpolation or statistical imputation methods that rely solely on temporal continuity, the LCA incorporates physical interdependencies observed in PV system behaviour, thereby enhancing the accuracy of imputed values.

Two linear regression models were developed based on historical data to characterize the relationship between irradiance and PV solar power, and module temperature and PV solar power. The corresponding scatter plots and regression lines are shown in Fig. 3 and Fig. 4, respectively. The derived empirical equations are expressed as follow:

$$G = \frac{P_{ac} - 194.48}{22.308} \quad (1)$$

$$T = \frac{P_{ac} + 13584}{2581.94} \quad (2)$$

These equations were derived by rearranging the original linear regression expressions to isolate the dependent variables, irradiance and temperature for the purpose of imputation. The performance of the fitted models is validated through their high coefficients of determination R^2 = 0.9283 for irradiance–solar power and R^2 = 0.896 for temperature–solar power. These high R^2 values confirm that the linear models reliably capture the underlying correlations, justifying their use for imputation.

To further evaluate the reliability of LCA imputation, a pseudo-missing test was conducted by removing data from 11:15 am to 7.25pm on 12 August 2022. The imputed values were then compared against the original observations. As shown in Fig. 5, For irradiance, the imputed irradiance and temperature closely followed the actual curves. Quantitatively, the imputation achieved RMSE = 50.30 W/m² (19.3%) and MAE = 29.00 W/m² (11.1%) for irradiance, and RMSE = 6.79 °C (20.1%) and MAE = 4.60 °C (13.6%) for temperature. These results confirm that the LCA method can reliably reconstruct both short and extended gaps without introducing significant bias into the dataset.

Apart from addressing missing data through imputation, detecting and removing outliers is essential for enhancing the accuracy and reliability of predictive models. Outliers can distort the learning process, causing models to capture noise rather than meaningful patterns, which increases the risk of overfitting. To mitigate this issue, the Interquartile Range (IQR) method is employed to effectively identify and eliminate outliers [21].

The IQR method is a widely used statistical approach that detects data points falling outside a specified range, typically 1.5 times the IQR above the third quartile (Q3) or below the first quartile (Q1) in order to ensure that the dataset is free from extreme values before being fed into the forecasting model [21].

The steps involved in the IQR method are as follows, and the process is visually represented in Fig. 6:

Step 1: Calculate the IQR.

$$IQR = Q3 - Q1 \quad (3)$$

Step 2: Determine the IQR threshold.

$$Threshold = 1.5 \times IQR \quad (4)$$

Step 3: Define the lower limit of outliers.

$$Lower\ limit = Q1 - (1.5 \times IQR) \quad (5)$$

Any values below this limit are classified as outliers and are removed.

Step 4: Define the upper limit of outliers.

$$Upper\ limit = Q3 + (1.5 \times IQR) \quad (6)$$

Similarly, any values above this limit are classified as outliers and are removed.

In this study, the IQR method was applied individually to solar irradiance, module temperature, and solar power data. To preserve natural diurnal characteristics, the procedure was restricted to daytime hours between 7:00 AM and 7:00 PM, ensuring that only abnormal daytime outliers were removed. Nighttime zero-power values were retained, as they reflect the natural operation of the PV system.

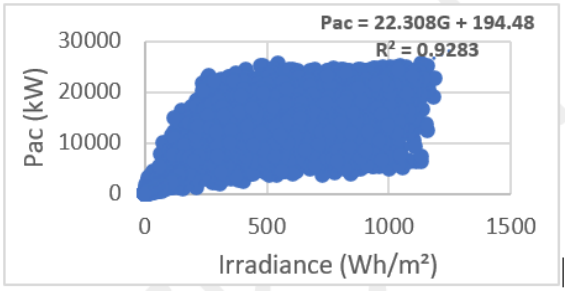

Fig. 3 Irradiance-solar power linear correlation approach.

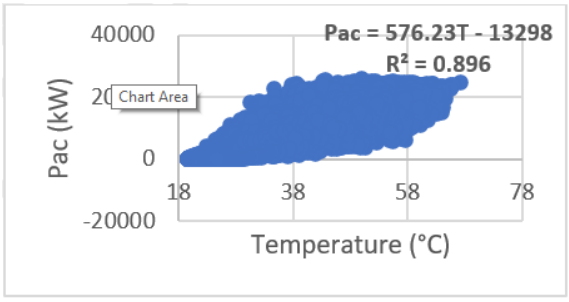

Fig. 4 Temperature-solar power linear approach.

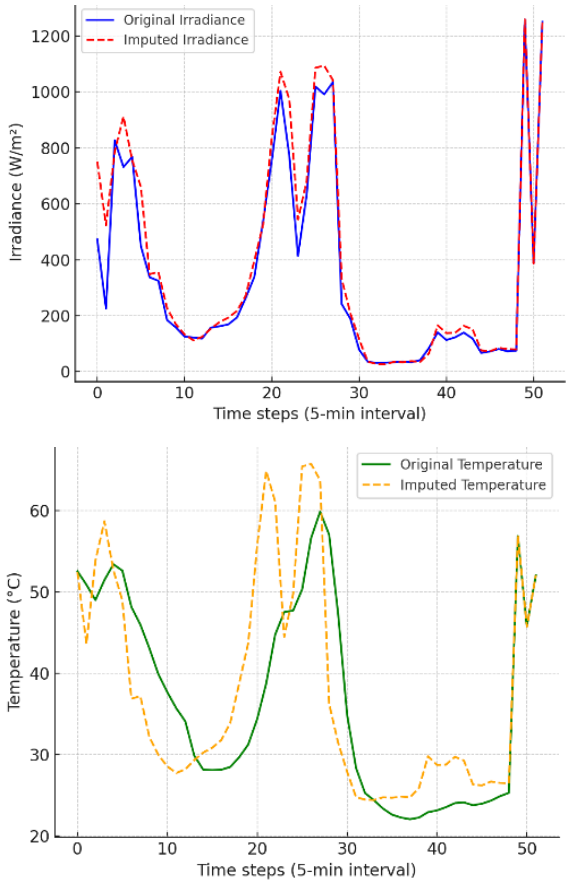

Fig. 5 Validation of the LCA.

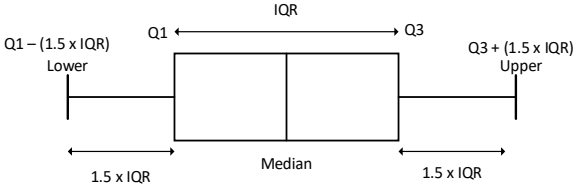

Fig. 6 IQR Technique for Identify Outliers.

4.3 Multivariate Sliding Window

The sliding window technique involves transforming time series data into a format suitable for supervised learning. This technique involves segmenting the continuous time series data into overlapping windows which each window contains fixed number of past observations (window size) which serve as input for forecasting model. By converting the time series data into a supervised learning form, the sliding window technique enables the model to learn from both the temporal dynamics and trend of the data.

This study employs a multivariate dataset comprising three input variables: solar irradiance, module temperature, and AC solar power to forecast AC solar power output. The window size is defined by the temporal resolution of the data. For example, at a 5-minute recorded interval, a 1-hour window consists of 12 steps, derived as 60 minutes divided by 5 minutes intervals. The sliding window method can be expressed mathematically to model temporal dependencies in multivariate time series data. Given window size *W,* the input sequence at time *t,* is defined as equation (7).

$$X(t) = \{X_G(t-k), X_T(t-k), X_P(t-k) | k = 0,1,\dots,W-1\} \quad (7)$$

Here, *k* denotes the lag index. X_G, X_T, and X_P represent solar irradiance, module temperature, and historical solar power, respectively. This sequence contains all observations from time *t* to *t - (W-1)*. Based on this input, the model forecasts the target solar power at a future horizon, *h* as defined in equation (8).

$$Y_P(t+h), \qquad h-1,2,\dots.,H \quad (8)$$

Where, *h* denotes the forecasting horizon (e.g, *h*=12 corresponds to a 1-hour ahead forecast for data sampled at 5-minute intervals.

Following each forecast, the sliding window advances one step, updating the indices to (*t+1)*. The updated input sequence is described in equation (9).

$$X(t+1) = \begin{Bmatrix} X_G(t+1-k), X_T(t+1-k), \\ X_P(t+1-k) \end{Bmatrix} \Big| \begin{matrix} k = 0,1,\dots, \\ W-1 \end{matrix} \quad (9)$$

As shown in Fig. 7, the forecasting process starts with "Input Sequence 1", encompassing the range *(t)* to (*t-11)* to forecast Y_P for *(t+12)*. Subsequently, the window shifts forward by one step, creating "Input Sequence 2," which spans *(t+1)* to *(t-10)*, and forecasts Y_P for *(t+13*). This iterative procedure repeats until forecasting is completed for the full forecasting horizon.

To capture both short-term and long-term temporal dependencies in solar power forecasting, the sliding window approach is systematically applied with varying window sizes, W=12, 24, 36, 60, 72, 144 and 288 steps. Smaller window sizes, such as 12 and 24 steps, focus on rapid fluctuations and short-term patterns, while larger sizes, such as 144 and 288 steps, are designed to capture long-term trends and seasonal variations. The model is trained and validated for each window size to ensure optimal predictive performance, achieved through iterative experimentation and accuracy evaluation across configurations.

4.4 Deep Learning Model Development

The fourth stage of this study focuses on the development and training of DL models for solar power forecasting. To ensure a fair and impartial comparison, the training is conducted using the same category of DL models specifically, LSTM, CNN and GRU. These models were strategically chosen due to their proven effectiveness in handling time-series data [22] and their ability to extract both sequential and spatial-temporal features from complex datasets [25-26].

LSTM and GRU network are advanced DL architectures designed to overcome the long-term memory limitations of traditional RNN. These architectures incorporate interconnected units equipped with gating mechanisms that control the flow of information. These gates enable the models to selectively retain or forget information as needed, ensuring effective handling of long-term dependencies in sequential data. Fig. 8 shows the architecture of an LSTM network consists of three gates namely forget, input and output gate. The forget gate determines which information from the cell state should be removed and to what extent. The input gate regulates the flow of new information into the cell state and decide to what extent new information should be incorporated in the current time step. The output gate extracts relevant information from the current input and updated the cell. The mathematical formulation used in the LSTM network is outlined in Equations (10) to (15) [24].

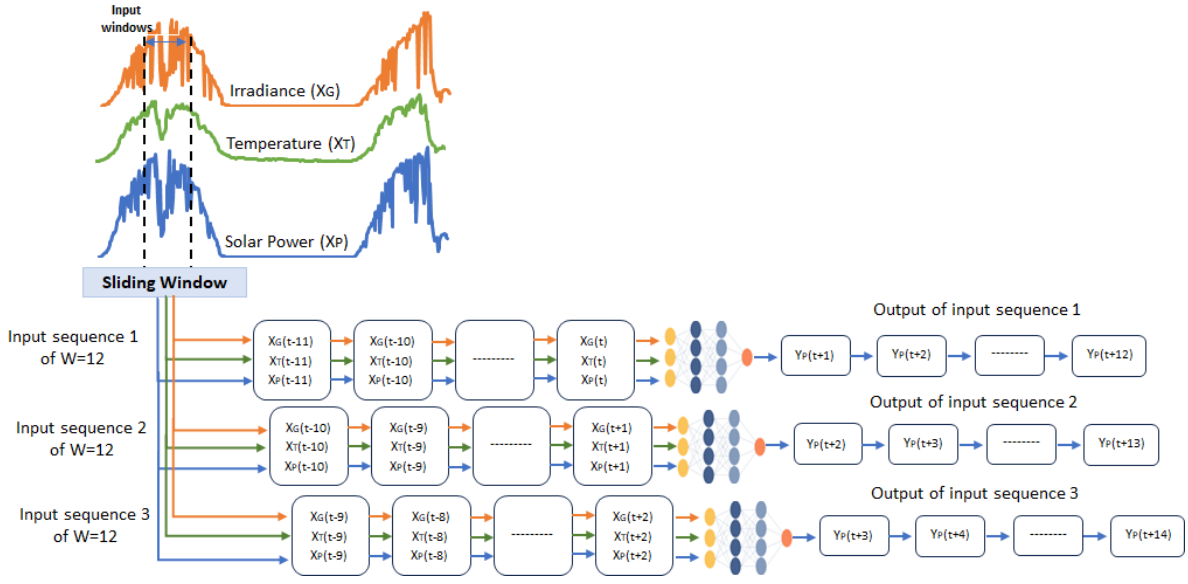

Fig. 7 Sliding window conceptual view for 1 hour ahead of solar power forecasting with 12 window size (W=12).

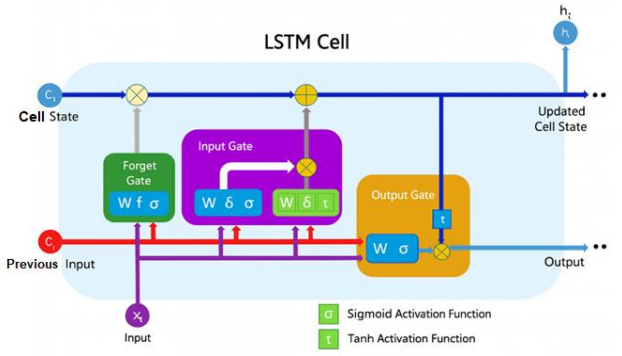

Fig. 8. LSTM architecture.

Equation (10) were utilised to compute forget gate units:

$$f_t = \sigma(W_f \times [h_{t-1}, x_t] + b_f \quad (10)$$

Equations (11) to (13) were utilised to update the input gate units:

$$i_t = \sigma(W_i \times [h_{t-1}, x_t] + b_i \quad (11)$$

$$\tilde{C_t} = tanh(W_C \times [h_{t-1}, x_t] + b_C \quad (12)$$

$$C_t = f_t(C_{t-1}) + i_t\tilde{C_t} \quad (13)$$

Equations (14) and (15) were utilised to compute output of the LSTM prediction:

$$o_t = \sigma(W_o \times [h_{t-1}, x_t] + b_o \quad (14)$$

$$h_t = o_t \times \tanh C_t \quad (15)$$

Here, f_t, i_t and o_t represent the forget, input and output gate, respectively, σ is the sigmoid activation function, W_f, W_i and W_o is the weight matrix for the forget, input and output gate, respectively. h_{t-1} is the hidden state from the previous time step x_t, is the input of the current time step and b_f, b_i and b_o are offset values for the forget, input and output gate, respectively.

The cell state, C_t is updated by integrating forget gate output f_t, the previous state C_{t-1} and the candidate cell state $\tilde{C_t}$. The candidate cell state is generated using a tanh activation function, with its value determined by the weight matrix W_c, and bias value b_c, which are specific to the candidate state computation. The updated C_t is processed through a tanh activation function to scale the values between -1 and 1, ensuring compatibility for subsequent updates. Meanwhile, the hidden state h_t, updated by the output gate, serves as the output for the current time step and as input for the next time step.

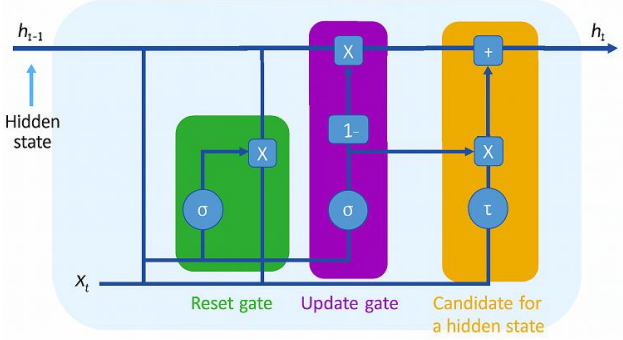

Fig. 9. GRU architecture.

Similar to LSTM, GRU also incorporate a gating mechanism, enabling them to capture long-term dependencies in sequential data. However, their architecture is simpler than LSTM as they have two gates: reset and update gate, as illustrated in Fig. 9. This fewer in parameters makes GRUs more computationally efficient [25].

The reset gate determines how much of the previous hidden state should be forgotten. It takes the previous hidden state h_{t-1}, and the current input x_t, to generate reset gate r_t. The reset gate r_t and update gate z_t are computed using equation (16) and (17) [26].

$$r_t = \sigma(W_r \times [h_{t-1}, x_t] + b_r \quad (16)$$

$$z_t = \sigma(W_z \times [h_{t-1}, x_t] + b_z \quad (17)$$

Here, σ represent the sigmoid activation function, W_r and W_z are the weight matrix for the reset gate and update gate, respectively. Similarly, b_r and b_z are their corresponding bias values.

The update gate regulates the contribution of the candidate activation vector $\tilde{h}_t$, to the new hidden state h_t. Using the previous hidden state, h_{t-1} and the current input x_t, its generates a vector of values ranging from 0 to 1. The candidate activation vector $\tilde{h}_t$, is "reset" by reset gate r_t, and computed using equation (18). Here, Wh is the weight matrix for the candidate hidden state and bh is its bias value.

$$\tilde{h}_t = tanh(W_h \times [r_t \times h_{t-1}, x_t] + b_h \quad (18)$$

Finally, the updated hidden state ht, is then computed as a weighted combination of the previous hidden state and the candidate activation vector, controlled by the update gate zt, using equation (19)

$$h_t = (1 - z_t) \times h_{t-1} + z_t \times \tilde{h} \quad (19)$$

CNN network is another DL model capable of handling complex and abstract features from data [30-31]. Although widely known for image processing, CNNs can also handle time series data with multiple variables, making them ideal for forecasting tasks. The standard CNN architecture, illustrated in Fig. 10, consists of several key layers: the input layer, convolutional layer, pooling layer, flattening layer, and a fully connected layer. The input layer receives pre-processed time-series data and prepares it for further analysis. The convolutional layer, the core component of the CNN, applies filters to scan the data and extract spatial features, enabling the detection of patterns in time-series data. Following the convolutional layer, the pooling layer reduces the dimensions of the feature maps by retaining the most important information, resulting in a smaller and manageable matrix. After pooling, the simplified feature maps are flattened into a single long vector and finally passed to the fully connected layer to perform the forecasting task.

4.5 Hyperparameter Tuning

The performance of DL models is highly sensitive to hyperparameter settings. In this study, the following hyperparameters were considered: number of CNN filters, kernel size, pooling size, stride length, number of LSTM units, dropout rate, learning rate, optimizer type, activation functions, epochs, and batch size. These represent the core parameters governing model architecture, training dynamics, and regularization. Failure to tune these parameters will lead to issues such as overfitting or underfitting during the training phase. Despite their importance, there is no fixed rule for setting the optimal hyperparameters for a model.

Various hyperparameter tuning have been proposed in the literature, including grid search [28], random search [29], and Bayesian optimisation [30]. However, in this study, hyperparameter tuning was conducted manually. While this approach can be time-consuming and requires iterative intervention, it offers valuable intuitive insights into the influence of individual hyperparameters on model performance. A manual tuning strategy was particularly suitable given the dataset size and computational constraints. The dataset covered nine months of 5-minute resolution data from a 25 MW utility-scale PV system, which already imposed substantial training demands. Employing automated search methods would have further increased computational costs exponentially, making them impractical within the available resources. Moreover, manual tuning reduced the risk of overfitting to a single validation subset and ensured that the models achieved stable and robust performance across different sliding window sizes and forecasting horizons.

The configuration set of parameter values employed during the tuning process is comprehensively presented in Table 1 and an overview of the optimal parameters identified for LSTM, GRU, and CNN model are outlined in Table 2.

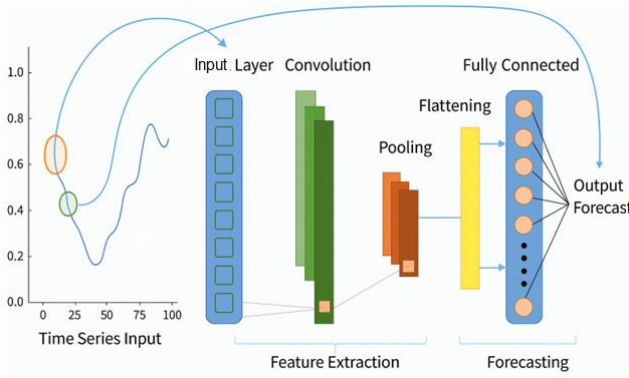

Fig. 10. CNN architecture.

Table 1 Configuration of model hyperparameter.

Parameter	LSTM	GRU	CNN
Neuron/Convolution	10, 30, 40, 50, 80, 100, 150, 200, 250, 300, 500	10, 30, 40, 50, 80, 100, 150, 200, 250, 300, 500	16, 32, 64, 128, 256, 512
Learning Rate	0.0001, 0.0005, 0.001, 0.005, 0.01, 0.05	0.0001, 0.0005, 0.001, 0.005, 0.01, 0.05	0.0001, 0.0005, 0.001, 0.005, 0.01, 0.05
Batch Size	16, 32, 64, 128, 256	16, 32, 64, 128, 256	16, 32, 64, 128, 256
Dropout	0, 0.1, 0.2, 0.3, 0.5	0, 0.1, 0.2, 0.3, 0.5	0, 0.1, 0.2, 0.3, 0.5
Optimizer	Adam, Adagrad, Nadam, Adamax, RMSprop	Adam, Adagrad, Nadam, Adamax, RMSprop	Adam, Adagrad, Nadam, Adamax, RMSprop
Epoch	30, 40, 50, 70, 100, 200	30, 40, 50, 70, 100, 200	30, 40, 50, 70, 100, 150, 200
Filter	-	-	2, 3, 4
Pool Size	-	-	2, 3, 4
Strides	-	-	2, 3

Table 2 Summarize the optimal model hyperparameter.

Parameter	LSTM	GRU	CNN
Neuron/Convolution	50	250	512
Learning Rate	0.001	0.001	0.001
Batch Size	32	128	128
Dropout	0.1	0	0.2
Optimizer	Adam	Adam	Adam
Epoch	100	100	150
Filter	-	-	3

4.6 Model Evaluation

The performance of the developed DL models in this study is evaluated using a combination of statistical validation and error-based evaluation metrics. The primary objective of this evaluation is to quantitatively assess how accurately each model forecast solar power output in comparison to actual observed values. Statistical validation is used to measure the strength of the relationship between forecasted and actual observed values and is most commonly conducted using the coefficient of determination (R^2) [31].

In parallel with statistical validation, various error measures are utilized to quantify the accuracy of forecasting. Among the widely used metrics in forecasting include Mean Absolute Error (MAE), Root Mean Squared Error (RMSE), Normalized Root Mean Squared Error (NRMSE), Mean Bias Error (MBE), Normalized Mean Bias Error (NMBE) and Mean Absolute Percentage Error (MAPE) [32].

Each error metric possesses distinct characteristics and serves specific analytical purposes. For instance, RMSE, MAE and MBE are scale-dependent metrics, meaning their values are influenced by the absolute magnitude of the dataset. MAPE is widely employed in forecasting to evaluate prediction accuracy. However, it is unsuitable for solar power forecasting. This limitation arises due to the occurrence of zero values during nighttime hours, which can lead to undefined or infinite MAPE values, thereby distorting model evaluation [33-34]. Considering these factors, this study adopts RMSE, MAE, and R^2 as the primary performance metrics for evaluating the DL forecasting models.

$$RMSE = \sqrt{\frac{\sum_1^N (y_t - \bar{y_t})^2}{N}} \quad (20)$$

$$MAE = \sum_1^N \left| \frac{y_t - \bar{y_t}}{N} \right| \quad (21)$$

$$R^2 = \frac{\sum_1^N (y_t - \bar{y_t})^2}{\sum_1^N (y_t - \hat{y_t})^2} \quad (22)$$

Where, y_t and $\bar{y_t}$ is the actual observed and forecasted value at time *t*, respectively and *N* is the number of observations.

5. Results and Analysis

This section examines the results pertaining to the optimal sliding window configuration for solar power forecasting across forecast horizons ranging from 1 to 12 hours. This analysis analysed 5-minute interval data collected from USPV system. In order to ensure thorough validation, the dataset was divided into two subsets: 70% for training and 30% for testing, with the training data covering the period from 1 August 2022 to 5 February 2023, and the test data from 6 to 28 February 2023.

The model was developed and trained using the Python programming language in the Jupyter Notebook environment, leveraging the TensorFlow and Keras frameworks to perform DL model. The data preprocessing and analysis process is supported by the Sci-kit Learn library and several other key Python libraries. The computing platform used consists of a Windows 10 operating system, powered by an Intel® Core™ i7-8750 @ 2.20 GHz processor, 16 GB RAM, and an NVIDIA GeForce GTX 1050 graphics processing unit (GPU), which provides sufficient computing resources to effectively perform training and inference models.

5.1 Performance Analysis of Sliding Window Approach

This study evaluates the performance of three DL models, in forecasting solar power for short-term and long-term horizons. To analyze the effect of input size on forecast accuracy, sliding window sizes ranging from 12 to 288 steps were utilized. The finding results are visualized in Fig. 11, 12, and 13. Specifically, Fig. 11 highlights RMSE evaluation by window size, Fig. 12 provides an analysis of MAE, and finally, Fig. 13 present R^2 correlations, illustrating the strength of the relationship between forecasted and actual solar power values for each model.

The results demonstrate that the sliding window size has a significant impact on the ability of DL models to capture temporal dependencies and produce accurate forecasts. Configurations with smaller input windows, such as 12-step and 24-step, exhibit poorer performance, with higher RMSE and MAE values and lower R^2 scores. This is primarily due to the insufficient historical data provided by smaller windows, which limits the model's ability to effectively capture temporal patterns. On the other hand, larger input windows, such as 72-step and 144-step configurations, improve forecasting performance by offering extensive historical data for the models to learn.

From Fig. 11, 12 and 13, the LSTM model shows consistent improvement with larger window sizes. Metrics such as RMSE and MAE showed a gradual decline, signifying enhanced accuracy, while R^2 values increased, highlighting improved alignment between forecast and actual values. However, when the window size exceeded 144 steps, the improvements showed slight declines, suggesting the model faced diminishing returns and the risk of overfitting. The LSTM model demonstrates superior performance with optimal window sizes of 144 steps. The model achieves its lowest RMSE with a 144-step window, recording values of 1,834 kW for 1-hour forecasting and 3,269 kW for 12-hour forecasting. Similarly, the lowest MAE values are observed at 1,022 kW for 1-hour forecast and 6,683 kW for 12-hour forecast. The R^2 scores for LSTM also peak at the 144-step window, showcasing its robustness in leveraging temporal dependencies.

The GRU model exhibited similar trends, with RMSE and MAE decreasing as the window size increased. The GRU model demonstrates strong performance for short-term forecasting horizons, when using smaller window sizes between 36 and 144 steps. GRU also achieved its best performance at a window size of 144 steps, where it balanced prediction accuracy and generalization capability. However, GRU showed slightly higher errors compared to LSTM for window size 144-step and above, suggesting it is marginally less effective in handling extensive temporal sequences, although it remains computationally efficient.

Although the CNN model was initially presumed to perform best with smaller or mid-range window sizes, owing to its architectural emphasis on local patterns, its optimal performance was observed at a 144-step window. At this optimal point, the model achieved an RMSE of 1,832 kW for a 1-hour forecast and a minimum MAE of 1,060 kW, proving its adaptability to longer temporal horizons. These results demonstrate the CNN's ability to adapt to longer temporal horizons, achieving competitive RMSE, MAE, and R^2 values compared to LSTM and GRU. For excessively large windows, however, CNN performance began to deteriorate, likely due to its reduced capacity to manage extended temporal contexts. Nonetheless, performance began to deteriorate with excessively large windows, likely due to its reduced efficiency in handling extended temporal dependencies.

Table 3 presents the training time and memory usage of the three models across different sliding window sizes. The results indicate that both computational time and memory requirements increase as the window size expands, owing to higher input dimensionality and model complexity. LSTM scales smoothly, maintaining relatively low training times, while GRU fluctuates but generally requires moderate resources, on the on the hand, CNN models generally required substantially longer training time and higher memory consumption.

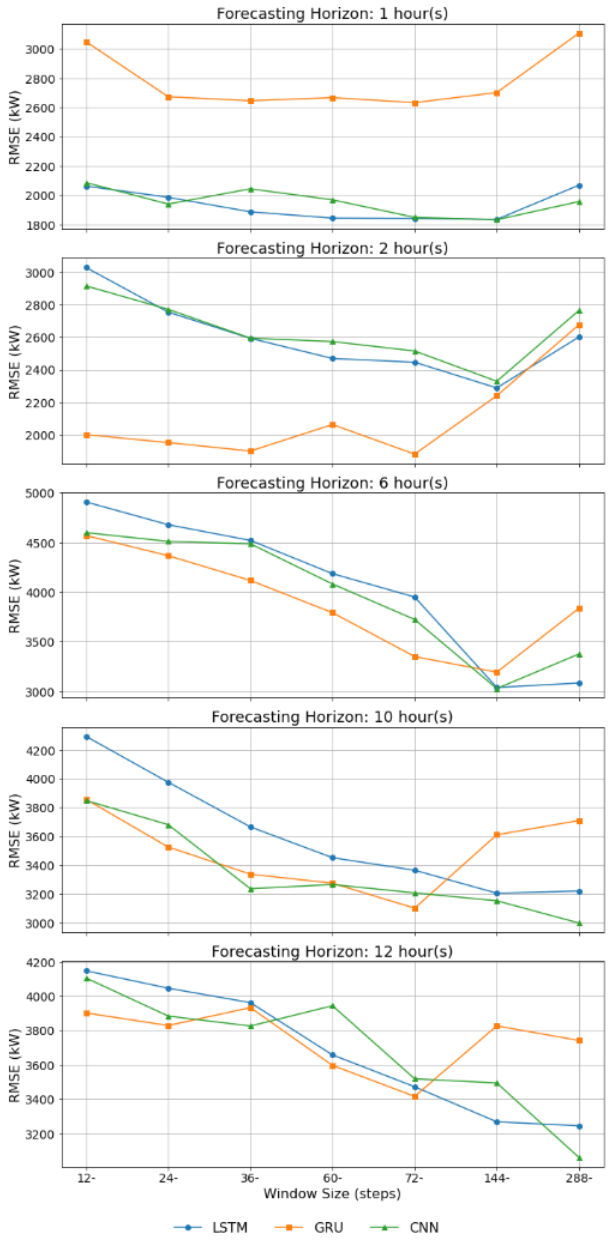

Fig. 11 RMSE evaluation across DL model and forecasting horizons at various window sizes.

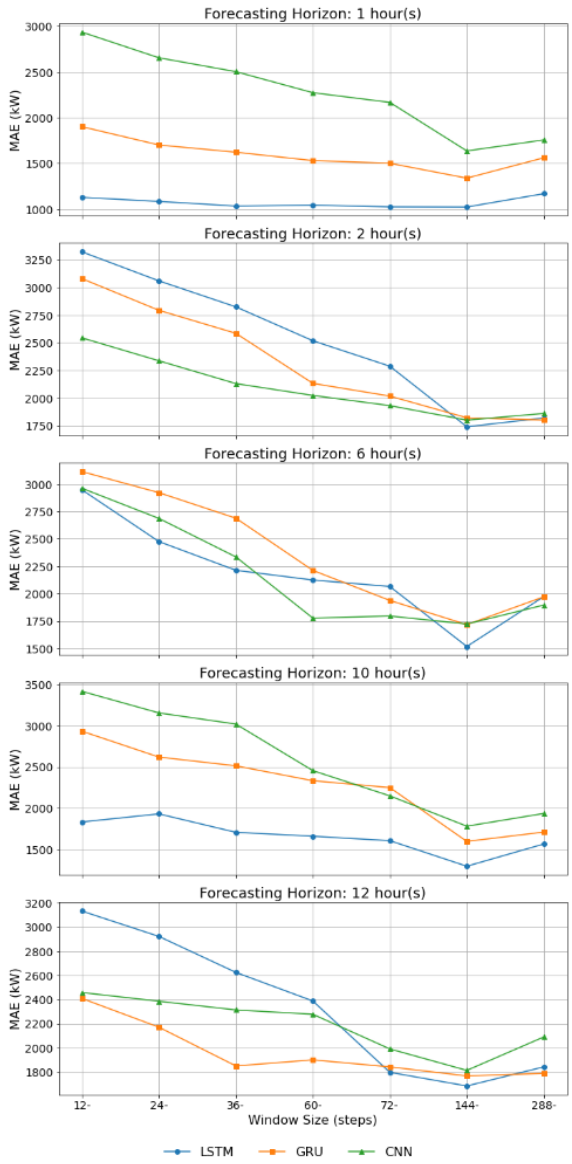

Fig. 12 MAE evaluation across DL model and forecasting horizons at various window sizes.

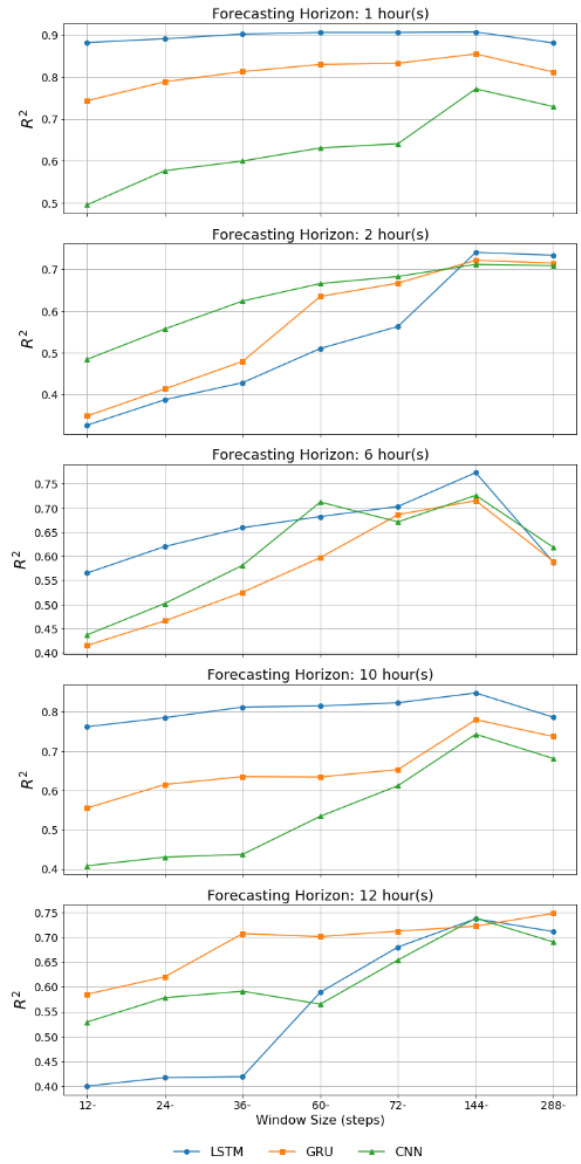

Fig. 13 R^2 evaluation across DL model and forecasting horizons at various window sizes.

Table 3 Training time and memory usage across different sliding window sizes.

Window Size	LSTM		CNN		GRU	
	Training Time (min)	Memory Usage (MB)	Training Time (min)	Memory Usage (MB)	Training Time (min)	Memory Usage (MB)
12	2.15	30.87	6.61	72.11	4.92	25.54
24	2.92	181.66	7.95	96.61	4.86	182.2
36	3.05	252.12	8.63	128.07	11.13	297.38
60	3.25	320.59	16.81	365.41	9.93	174.84
72	3.72	370.46	17.73	300.48	8.45	125.52
144	4.76	176.59	32.62	513.82	9.26	467.39
288	10.74	774.95	89.76	538.05	10.29	484.84

While the 288-step window provides the longest historical context, the size does not always provide further improvements and occasionally shows slight performance degradation, potentially due to overfitting or diminishing returns on additional historical information. At the same time, it significantly raises the computational cost. For instance, at *W*=288, CNN required 89.76 minutes and 538 MB of memory, compared to 10.74 minutes and 774.95 MB for LSTM, and 10.29 minutes and 484.84 MB for GRU. Considering both forecasting performance and computational efficiency, the 144-step window emerged as the most practical choice. At this size, the models delivering strong forecasting accuracy while maintaining moderate training time and memory usage across models.

5.2 Comparative Evaluation of Deep Learning Models at the Optimal Window Size

Based on the findings in Section 5.1, the optimal window sizes of 144 steps for all models are utilised in this section. The box plots in Fig. 14 provides a comparative visualization of the models performance across RMSE, MAE, and R^2 metrics. LSTM demonstrated superior performance, achieving the lowest mean RMSE of 2,727.2 kW and MAE of 1,452.6 kW, highlighting its superior ability to minimize both large and average forecasting errors. Additionally, it recorded the highest mean R^2 of 0.801, reflecting a strong correlation between forecasted and actual solar power values. GRU, despite its simpler architecture, however, delivers competitive results. GRU achieved a mean RMSE of 3,114.4 kW and a mean MAE of 1,648.4 kW which both slightly higher than LSTM metrics. GRU mean R^2 of 0.759 is also slightly

lower than LSTM, indicating a slightly weaker correlation compared to LSTM. CNN, on the other hand, showed limitations in capturing long-term patterns. While its mean RMSE of 2767.4 kW which is closer to LSTM, its mean MAE of 1751 kW is the highest among the three models, reflecting less precision in minimizing average errors. Moreover, CNN achieves the lowest mean R^2 of 0.738, indicating a weaker correlation between its predictions and actual values. These findings suggest that CNN struggles with variability and consistency, particularly when handling longer temporal horizons. Among all the three models, the results clearly indicate that at the optimal window size, LSTM consistently outperforms GRU and CNN across all evaluation metrics, making it the most reliable model for this forecasting task.

Table 4 presents the percentage improvement of forecasting using optimal window size compared to one-step window size. The LSTM model delivers consistent improvements across all metrics, with greater performance gains at 60-minute forecasting horizons. For example, RMSE improves from 3.4% at 10 minutes to 23.1% at 60 minutes. MAE follows a similar trend, improving from 3.3% to 30.7%, reflecting the model's proficiency in reducing forecasting errors. The R^2 metric also improves steadily, with an increase from 0.1% at 10 minutes to 8.6% at 60 minutes, demonstrating stronger correlations between forecasts and actual values as the optimal sliding window size helps LSTM capture temporal dependencies more effectively.

The CNN model excels in improving RMSE and MAE, particularly over mid-to-longer forecasting horizons. For RMSE, CNN surpasses GRU with a 33.4% improvement at the 30-minute horizon and consistent progress peaking at 27.9% at 60 minutes. The R^2 metric reflects steady growth, from 0.3% at 10 minutes to 10.9% at 60 minutes, indicating CNN's increasing effectiveness in capturing temporal patterns over extended forecasting windows.

On average, LSTM enhanced by 14.4% in RMSE, 19.3% in MAE, and 3.18% in R2. The GRU model exhibited an average improvement of 15.4% in RMSE, 26.2% in MAE, and 2.3% in R2. CNN exhibited the highest average improvement of 25.4%, 22.4%, and 4.3% in RMSE, MAE, and R2, respectively. The results demonstrate that the optimal sliding window size significantly enhances the performance of all models compared to a one-step window size. These findings highlight the importance of tailoring the sliding window size to the model architecture and forecasting requirements

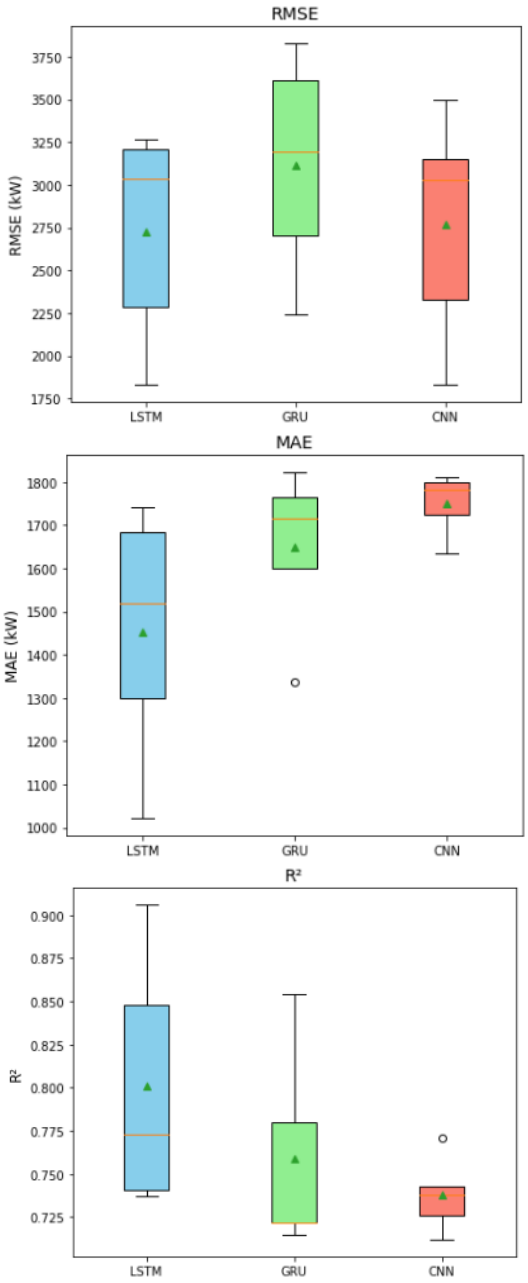

Fig. 14 RMSE, MAE and R^2 evaluation at optimal window size of 144 steps.

Table 4 Improvement of optimal sliding window size forecasting.

Model	Evaluation	One-step Windows size						Optimal Windows size						Improvement (%)					
		10 min	20 min	30 min	40 min	50 min	60 min	10 min	20 min	30 min	40 min	50 min	60 min	10 min	20 min	30 min	40 min	50 min	60 min
LSTM	RMSE	587	1004	1397	1764	2092	2431	568	891	1158	1449	1671	1869	3.4	11.3	17.1	17.9	20.1	23.1
	MAE	317	540	773	1004	1222	1438	307	463	605	747	895	996	3.3	14.3	21.7	25.6	26.8	30.7
	R^2	0.99	0.97	0.945	0.91	0.877	0.834	0.99	0.97	0.96	0.94	0.92	0.90	0.1	0.6	1.8	3.1	5.1	8.6
GRU	RMSE	533	966	1350	1756	2057	2389	438	806	1180	1467	1749	2063	17.8	16.6	12.6	16.5	15.0	13.6
	MAE	306	552	746	1016	1283	1399	213	422	576	744	883	1085	30.4	23.6	22.8	26.8	31.2	22.4
	R^2	0.99	0.97	0.94	0.91	0.881	0.84	0.99	0.98	0.96	0.94	0.91	0.88	0.3	0.8	1.3	2.8	3.7	4.9
CNN	RMSE	557	974	1504	1791	2038	2577	461	721	1001	1356	1562	1860	17.2	26.0	33.4	24.3	23.4	27.9
	MAE	338	538	820	993	1175	1561	250	439	629	841	947	1060	26.0	18.4	23.3	15.3	19.4	32.1
	R^2	0.991	0.973	0.93	0.91	0.884	0.814	0.994	0.985	0.972	0.948	0.932	0.903	0.3	1.2	3.7	4.2	5.4	10.9

6. Conclusion

This paper has presented a DL model approach for solar power forecasting of USPV systems. This study contributes to the development and evaluation of an optimal sliding window approach to capture temporal dependencies across multiple forecasting horizons effectively. The analysis identifies an ideal window size of 144 steps, resulting in notable accuracy enhancements compared to conventional one-step forecasting approaches. Among the evaluated models, LSTM demonstrates the highest forecasting accuracy across all horizons, attributed to its sophisticated gating mechanism that effectively captures long-range temporal dependencies. Future work could explore hybrid DL architectures to further optimize window selection, integrate automated optimization methods to reduce the limitations of manual tuning, and advance toward probabilistic forecasting to better quantify uncertainty. The proposed sliding window framework provides grid operators with actionable insights to improve scheduling, reduce mismatch losses, and enhance the stability of renewable energy integration.

Acknowledgements

The authors would like to acknowledge the Universiti Teknologi MARA (UiTM) and UiTM Energy & Facilities for providing data in completing this research.

References

[1] Wilson, G. M., Al-Jassim, M., Metzger, W. K., Glunz, S. W., Verlinden, P., Xiong, G., Mansfield, L. M., Stanbery, B. J., Zhu, K., Yan, Y., Berry, J. J., Ptak, A. J., Dimroth, F., Kayes, B. M., Tamboli, A. C., Peibst, R., Catchpole, K., Reese, M. O., Klinga, C. S., Denholm, P., Morjaria, M., Deceglie, M. G., Freeman, J. M., Mikofski, M. A., Jordan, D. C., TamizhMani, G. and Sulas-Kern, D. B., The 2020 photovoltaic technologies roadmap. *Journal of Physics D: Applied Physics*. 53 (2020) 493001, doi: https://doi.org/10.1088/1361-6463/ab9c6a.

[2] Zainuddin, H., Hanifah, M. M. M., Hussin, M. Z. and Muhammad, N. (2024). Cell temperature determination based on IEC61215: Solar photovoltaic experimental study of tropical Malaysia. *Journal of Renewable Energy and Smart Grid Technology*. 19 (2) 1–6, doi: https://doi.org/10.69650/rast.2024.258005.

[3] Energy Commission. *LSS Progress By Region (Q1 2024),* <https://www.st.gov.my/eng//web/general/details/999> (2024).

[4] Hammad, M., Khalil, S. and Mahmoud, I. M. Application of nonparametric ML on forecasting the production of a large-scale solar power plant: Kom-Ombo case study. *Sustainable Computing: Informatics and Systems*. 45 (2025) 101074.

[5] Thansouphanh, B., Premrudeepreechacharn, S., Srirattanachaikul, W. and Ngamsanroaj, K., Predicting the temperature trend of a generator using RNN, GRU, and LSTM algorithms at Nam Ngum 1 hydropower plant in Laos. *Journal of Renewable Energy and Smart Grid Technology.* 20 (2025) 54–60, doi: https://doi.org/10.69650/rast.2025.260236.

[6] Abumohsen, M., Owda, A. Y., Owda, M. and Abumihsan, A., Hybrid machine learning model combining of CNN-LSTM-RF for time series forecasting of solar power generation. e-Prime - Advances in Electrical Engineering. *Electronics and Energy*. 9 (2024) 100636, doi: https://doi.org/10.1016/j.prime.2024.100636.

[7] Liu, C. H., Gu, J. C. and Yang, M. T., A simplified LSTM neural networks for one day-ahead solar power forecasting. *IEEE Access*. 9 (2021) 17174–17195, doi: https://doi.org/10.1109/ACCESS.2021.3053638.

[8] Ozbek, A., Yildirim, A. and Bilgili, M., Deep learning approach for one-hour ahead forecasting of energy production in a solar-PV plant. *Energy Sources, Part A: Recovery, Utilization, and Environmental Effects. 44* (2022) 10173–10186, doi: https://doi.org/10.1080/15567036.2021.1924316.

[9] Kumari, P. and Toshniwal, D., Long short term memory–convolutional neural network based deep hybrid approach for solar irradiance forecasting. *Applied Energy.* 295 *(*2021) 117061, doi: https://doi.org/10.1016/j.apenergy.2021.117061.

[10] Bhatt, A., Ongsakul, W., Nimal Madhu, M. and Singh, J. G., Sliding window approach with first-order differencing for very short-term solar irradiance forecasting using deep learning models. *Sustainable Energy Technologies and Assessments*. 50 (2022) 101864, doi: https://doi.org/10.1016/j.seta.2021.101864.

[11] Cican, G., Buturache, A.-N. and Silivestru, V., Predicting photovoltaic energy production using neural networks: Renewable integration in Romania. *Processes*. 13 (2025) 2219, doi: https://doi.org/10.3390/pr13072219.

[12] du Plessis, A. A., Strauss, J. M. and Rix, A. J., Short-term solar power forecasting: Investigating the ability of deep learning models to capture low-level utility-scale photovoltaic system behaviour. *Applied Energy*. 285 (2021) 116395, doi: https://doi.org/10.1016/j.apenergy.2020.116395.

[13] Xu, Y., Zheng, S., Zhu, Q., Wong, K. C., Wang, X. and Lin, Q., A complementary fused method using GRU and XGBoost models for long-term solar energy hourly forecasting. *Expert Systems with Applications*. 254 (2024) 124286, doi: https://doi.org/10.1016/j.eswa.2024.124286.

[14] Kim, J., Kim, H., Kim, H. G., Lee, D. and Yoon, S., A comprehensive survey of deep learning for time series forecasting: Architectural diversity and open challenges. *Artificial Intelligence Review*. 58 (2025) 216, doi: https://doi.org/10.1007/s10462-025-11223-9.

[15] Huang, X., Shi, J., Gao, B., Tai, Y., Chen, Z. and Zhang, J., Time series forecasting for hourly photovoltaic power using conditional generative adversarial network and Bi-LSTM. *Energy*. 246 (2022) 123403, doi: https://doi.org/10.1016/j.energy.2022.123403.

[16] Boubaker, S., Benghanem, M., Mellit, A., Lefza, A., Kahouli, O. and Kolsi, L., Deep neural networks for predicting solar radiation at Hail Region, Saudi Arabia. *IEEE Access*. 9 (2021) 36719–36729, doi: https://doi.org/10.1109/ACCESS.2021.3062205.

[17] Yang, D., Li, M., Guo, J. and Du, P., An attention-based multi-input LSTM with sliding window-based two-stage decomposition for wind speed forecasting. *Applied Energy*. 375 (2024) 124057, doi: https://doi.org/10.1016/j.apenergy.2024.124057.

[18] Cui, X., Yu, X., Niu, H., Niu, D. and Liu, D., A novel data-driven multi-step wind power point-interval prediction framework integrating sliding window-based two-layer adaptive decomposition and multi-objective optimization for balancing prediction accuracy and stability. *Applied Energy*. 397 (2025) 126348, doi: https://doi.org/10.1016/j.apenergy.2025.126348.

[19] Rana, M. and Rahman, A., Multiple steps ahead solar photovoltaic power forecasting based on univariate machine learning models and data re-sampling. *Sustainable Energy, Grids and Networks.* 21 (2020) 100286, doi: https://doi.org/10.1016/j.segan.2019.100286.

[20] Sharadga, H., Hajimirza, S. and Balog, R. S., Time series forecasting of solar power generation for large-scale photovoltaic plants. *Renewable Energy*. 150 (2020) 797–807, doi: https://doi.org/10.1016/j.renene.2019.12.131.

[21] Schwertman, N. C., Owens, M. A. and Adnan, R., A simple more general boxplot method for identifying outliers. *Computational Statistics & Data Analysis.* 47 (2004) 165–174, doi: https://doi.org/10.1016/j.csda.2003.10.012.

[22] Xiao, Z., Huang, X., Liu, J., Li, C. and Tai, Y., A novel method based on time series ensemble model for hourly photovoltaic power prediction. *Energy.* 276 (2023) 127542, doi: https://doi.org/10.1016/j.energy.2023.127542.

[23] Wang, J., Guo, L., Zhang, C., Song, L., Duan, J. and Duan, L., Thermal power forecasting of solar power tower system by combining mechanism modeling and deep learning method. *Energy*. 208 (2020) 118403, doi: https://doi.org/10.1016/j.energy.2020.118403.

[24] Ahmed, R., Sreeram, V., Togneri, R., Datta, A. and Arif, M. D., Computationally expedient photovoltaic power forecasting: A LSTM ensemble method augmented with adaptive weighting and data segmentation technique. *Energy Conversion and Management*. 258 (2022) 115563, doi: https://doi.org/10.1016/j.enconman.2022.115563.

[25] Abumohsen, M., Owda, A. Y. and Owda, M., Electrical load forecasting using LSTM, GRU, and RNN algorithms. *Energies*. 16 (2023) 2283, doi: https://doi.org/10.3390/en16052283.

[26] Jia, P., Zhang, H., Liu, X. and Gong, X., Short-term photovoltaic power forecasting based on VMD and ISSA-GRU. *IEEE Access.* 9 (2021) 105939–105950, doi: https://doi.org/10.1109/ACCESS.2021.3099169.

[27] Kabir, S., Islam, R. U., Hossain, M. S. and Andersson, K., An integrated approach of belief rule base and convolutional neural network to monitor air quality in Shanghai. *Expert Systems with Applications*. 206 (2022) 117905, doi: https://doi.org/10.1016/j.eswa.2022.117905.

[28] Sulistya Kurniawan, A. D., Imanda Purnama, Y., Wicaksono, A. B. and Mahmudah, H. Comparative analysis and optimization of deep learning models for object detection using grid search hyperparameter tuning. in *2024 International Electronics Symposium (IES).* (2024), 587–592, doi: https://doi.org/10.1109/IES63037.2024.10665797.

[29] Quan, S. J., Comparing hyperparameter tuning methods in machine learning based urban building energy modeling: A study in Chicago. *Energy and Buildings*. 317 (2024) 114353, doi: https://doi.org/10.1016/j.enbuild.2024.114353.

[30] Passos, D. and Mishra, P., A tutorial on automatic hyperparameter tuning of deep spectral modelling for regression and classification tasks. *Chemometrics and Intelligent Laboratory Systems.* 223 (2022) 104520, doi: https://doi.org/10.1016/j.chemolab.2022.104520.

[31] Chicco, D., Warrens, M. J. and Jurman, G., The coefficient of determination R-squared is more informative than SMAPE, MAE, MAPE, MSE and RMSE in regression analysis evaluation. *PeerJ Computer Science*. 7 (2021) e623, doi: https://doi.org/10.7717/PEERJ-CS.623.

[32] Barhmi, K., Heynen, C., Golroodbari, S. and van Sark, W., A review of solar forecasting techniques and the role of artificial intelligence. *Solar*. 4 (2024) 101–139, doi: https://doi.org/10.3390/solar4010005.

[33] Mei, F., Gu, J., Pan, S., Zhong, L., Lu, J., Jiang, Y. and Zheng, J., Day-ahead nonparametric probabilistic forecasting of photovoltaic power generation based on the LSTM-QRA ensemble model. *IEEE Access*. 8 (2020) 166138–166149, doi: https://doi.org/10.1109/ACCESS.2020.3021581.

[34] Nguyen, N. Q., Bui, L. D., Doan, B. V., Sanseverino, E. R., Di Cara, D. and Nguyen, Q. D., A new method for forecasting energy output of a large-scale solar power plant based on long short-term memory networks: A case study in Vietnam. *Electric Power Systems Research*. 199 (2021) 107427, doi: https://doi.org/10.1016/j.epsr.2021.107427.