Journal of Applied Automation Technologies

Research Paper

# A Temporal-Consistency-Enhanced BiLSTM Method for Photovoltaic Power Forecasting in Distributed PV Stations

**Rasha Al-Qarni[1,*], Zuhaira Al-Buainain[1] and Obeida Al-Zabeidi[1]**

[1] Department of Computer Science and Engineering, American University of Sharjah, 26-666 Sharjah, United Arab Emirates
*Corresponding author: rasha.da@aus.edu

**Abstract.** Distributed photovoltaic stations are increasingly operated as dense, weather-sensitive generation clusters, yet their short-term power forecasts often contain discontinuous ramps that are inconsistent with plant physics and recent temporal evolution. This paper proposes a temporal-consistency-enhanced bidirectional long short-term memory method, termed TC-BiLSTM, for 15-min-ahead to 4-h-ahead photovoltaic power forecasting in distributed PV stations. The method integrates normalized station measurements, meteorological variables, solar-position descriptors, and historical ramp features into a two-stream BiLSTM encoder. A compact consistency module then constrains adjacent forecast horizons by combining ramp-aware smoothness, irradiance-response monotonicity, and capacity-normalized error terms. Experiments are organized on a multi-station dataset containing 38 distributed PV stations, 18 months of operation, 96 daily sampling points, and four weather categories. Compared with persistence, SVR, CNN-LSTM, vanilla LSTM, and standard BiLSTM baselines, TC-BiLSTM reduces RMSE from 8.74% to 6.58% of installed capacity, decreases MAE from 5.21% to 3.91%, and improves ramp-event F1-score from 0.704 to 0.823. The gain is most visible on partly cloudy days, where the average ramp error falls by 22.6%. The results indicate that temporal consistency is a practical regularization principle for distributed PV forecasting because it improves point accuracy while producing smoother and more dispatchable forecast trajectories.

**Keywords:** *BiLSTM, Temporal Consistency, Photovoltaic Power Forecasting, Distributed PV Stations, Renewable Energy Dispatch*

Received on 24 October 2023, Accepted on 19 July 2024, Published on 28 July 2024



## Introduction

Distributed photovoltaic generation has shifted the operating mode of distribution networks from a problem of load-following to a bidirectional and weather-coupled balancing problem. When a feeder with hundreds of rooftops and small ground-mounted photovoltaic units passes through an area at the edge of a cloud, its total output can suddenly change, and the same event may occur at nearby sites with only a few sampling intervals of delay [1]. Therefore, the forecast that operators require needs to be precise at certain times and also relatively stable throughout the entire scheduling period. Most forecasting services focus on point errors and consider the shape of the predicted curve to be a minor problem. This division is not convenient for actual dispatch; even if the mean error is relatively small, excessive reserve actions may be generated due to fluctuations in the forecast path from one period to the next [2]. Recently, data-driven photovoltaic power generation forecasts have improved non-linear feature extraction with deep recurrent and convolutional networks, but the temporal consistency of the output sequence is still not well defined [3]. The problem is more severe at distributed stations because the ratings of individual plants are small, measurement noise is relatively large, and weather sensors are not uniformly installed [4].

Short-term PV power forecasting generally uses numerical weather prediction, local irradiance data, historical power data and calendar or solar-position features. Although the input carries the necessary physical information to predict the output envelope, a data-driven model may still produce a local discontinuity if the

https://doi.org/10.64972/jaat.2024v2.316p26e:361-373

training objective is dominated by independent point-wise loss. A recurrent network can remember previous states, but it does not automatically need to respect the feasible ramp imposed by recent irradiance changes, installed capacity, inverter clipping and station-level inertia [5]. BiLSTM models are also restricted to this extent. Although the bidirectional structure can learn more context from the input window, the decoder's output at a given time step is relatively independent of other forecast steps if those horizons are not strongly coupled targets [6]. Distribution-system scheduling has introduced a new problem of temporal inconsistency in feeder power-flow calculations, storage dispatch and curtailment decisions, and all artificial oscillations must now be treated as possible physical ramps [7].

Develop a temporal-consistency-enhanced BiLSTM method for photovoltaic power forecasting of distributed PV stations in this paper. In short, a forecast needs to meet the measured goals and also move along the same path as the recent power curve and irradiance-driven ramp tendency over time. TC-BiLSTM is a bidirectional sequence encoder for temporal context extraction, a horizon-aware decoder for generating multi-step predictions, and a consistency regularizer that connects adjacent prediction points through slope, curvature, and irradiance response constraints. The form of the study is an engineering forecasting method, not a general neural network experiment. Set up a reproducible data pipeline, define the loss function, and evaluate the model in clear, partly cloudy, overcast and rainy conditions. The following sections introduce related work, define the temporal-consistency problem, present the forecasting architecture and analyze the results of multi-station distributed PV data.

## Related Work

### Data-Driven PV Power Forecasting

Statistical and machine-learning PV forecasting methods have gradually moved from single-plant regression to multi-source temporal models. Persistence and autoregressive baselines are still good reference values because they show the inherent difficulty of short horizons, especially in stable clear-sky periods [8]. Support Vector Regression, Random Forest and Gradient Boosting can enhance the mapping between weather variables and power output; however, their capacity for utilizing long sequential context is relatively weak unless significant feature engineering is employed [9]. Deep learning models directly learn nonlinear temporal features from the history to address this problem. CNN-based models are suitable for extracting local fluctuation patterns, and LSTM and gated recurrent units can learn long-term dependencies in the input window [10].

BiLSTM models are suitable for PV forecasting because they can process the input window in both directions chronologically. The backward pass does not have access to future forecast targets; instead, it can use the full observed history window for representation at each time step. Thus, it can distinguish a genuine ramp caused by clouds from random sensor noise. Most BiLSTM studies still use MAE, MSE or their weighted forms for training. The above loss function enhances point-level fitting but fails to directly constrain properties such as smoothness and curvature of the forecast curve.

### Temporal Modeling and Distributed PV Challenges

Several studies have attempted to improve renewable energy forecasting by adding attention mechanisms, decomposition, hybrid optimization, or probabilistic calibration. Attention mechanisms can give more weight to influential irradiance, humidity and lagged-power variables, and decomposition methods can separate high-frequency components before prediction [11]. Probabilistic Methods also determine the range of uncertainty and are applied to reserve scheduling and market bidding [12]. However, these methods often assess the accuracy of the forecast based on marginal errors or interval coverage, and the temporal coherence of the deterministic trajectory is only indirectly addressed.

Distributed PV stations also increase the complexity of model design through spatial diversity. Neighboring stations share the same weather system but have different orientations, inverter capacities, local shading, data quality and maintenance statuses [13]. A model that is too flexible may overfit the station-specific irregularities, and a model that is too smooth may fail to capture the real cloud-edge ramps. Thus, the deficiency is not merely the absence of non-linearity. It is the lack of an explicit training signal that informs a forecasting model when to

https://doi.org/10.64972/jaat.2024v2.316p26e:361-373

keep the adjacent predicted points close, when they may change rapidly, and how such changes should align with observable irradiance and historical ramp evidence [14].

## Problem Formulation and Temporal Consistency

### Distributed PV Forecasting Task

The forecasting problem is to predict the generation of a set of distributed PV stations at a fixed 15-minute sampling interval. For a station s and time index $t$, the available feature vector includes measured active power, global horizontal irradiance, plane-of-array irradiance (when available), ambient temperature, module temperature, humidity, wind speed, clear-sky index, solar elevation, time-of-day encoding, station capacity, and several lagged ramp descriptors. The Goal is a multi-step power sequence over the next H intervals. All power variables are normalised by installed capacity so that stations with different ratings can be trained in a shared model without being dominated by large plants in the loss.

Matrix X contains the input window, where the columns are normalized features and the rows are the most recent L time steps. The model outputs a vector of future capacity-normalized PV outputs. This construction maintains the compatibility of the forecasting problem with both station-level and aggregated feeder-level deployment by using the same target definition before or after station aggregation. The basic supervised map is written as follows.

$$X_t^s = [x_{t-L+1}^s, x_{t-L+2}^s, \dots, x_t^s] \quad \text{Eq.(1)}$$

Here the notation keeps all station outputs on a common capacity-normalized scale, which makes the loss comparable across heterogeneous distributed PV assets.

$$y_t^s = [p_{t+1}^s, p_{t+2}^s, \dots, p_{t+H}^s] \quad \text{Eq.(2)}$$

Here the notation keeps all station outputs on a common capacity-normalized scale, which makes the loss comparable across heterogeneous distributed PV assets.

$$p_t^s = P_t^s / C_s \quad \text{Eq.(3)}$$

Here the notation keeps all station outputs on a common capacity-normalized scale, which makes the loss comparable across heterogeneous distributed PV assets.

$$r_t^s = p_t^s - p_{t-1}^s \quad \text{Eq.(4)}$$

Here the notation keeps all station outputs on a common capacity-normalized scale, which makes the loss comparable across heterogeneous distributed PV assets.

A practical detail is that the same station can show different effective dynamics under different operating states. Inverter clipping, local curtailment, maintenance outages, and delayed meteorological telemetry are not part of the same forecasting mechanism as ordinary irradiance-to-power conversion. The proposed data pipeline therefore treats unavailable power, long missing weather segments, and abnormal constant-output periods as state information instead of silently mixing them with valid ramp samples. Short gaps below two sampling intervals are interpolated because they usually reflect communication delay, while longer gaps are masked during supervised training.

### Ramp-Aware Temporal Consistency

A ramp term has been added because distributed PV forecasts are frequently used in the dispatch tool, and such forecasts are more sensitive to changes in generation than fixed-output values. A forecast may have a small average error, but it is still difficult to use if its ramp sequence alternates signs in a nonphysical way. A clear-sky index is employed as a reduced irradiance descriptor. It separates the astronomical variation from the weather-induced attenuation and can thus learn that a power reduction near sunset is different from a cloud-shadow drop at noon.

Temporal consistency is used as a training condition instead of a post-processing filter. Post-processing can smooth the predicted curve, but it may also remove actual ramps, and it does not inform the encoder why a ramp should occur. TC-BiLSTM's loss function is a combination of point accuracy, a penalty for adjacent-horizon

https://doi.org/10.64972/jaat.2024v2.316p26e:361-373

ramp mismatch, and a second-order curvature term. The first term has good numerical accuracy, and the following terms prevent isolated spikes that are not due to changes in the input.

$$L_{acc} = (1/H)\sum_{h=1}^{H} \left|y_{t,h} - \hat{y}_{t,h}\right| \quad \text{Eq.(5)}$$

This term is evaluated along the forecast horizon so that adjacent predicted points are trained as a coupled sequence rather than as isolated regression outputs.

$$\Delta\hat{y}_{t,h} = \hat{y}_{t,h} - \hat{y}_{t,h-1} \quad \text{Eq.(6)}$$

This term is evaluated along the forecast horizon so that adjacent predicted points are trained as a coupled sequence rather than as isolated regression outputs.

$$L_{ramp} = (1/(H-1))\sum_{h=2}^{H} \left|\Delta y_{t,h} - \Delta\hat{y}_{t,h}\right| \quad \text{Eq.(7)}$$

This term is evaluated along the forecast horizon so that adjacent predicted points are trained as a coupled sequence rather than as isolated regression outputs.

The division is necessary because a distribution operator does not consume the forecast as disconnected scalar values. The operator sees a curve that adjusts reserve allocation, storage charging, voltage-control preparation, and sometimes feeder reconfiguration. When an artificial one-step spike occurs, downstream tools may be too flexible or incorrectly identify it as a cloud front. When a real ramp is over-smoothed, the same tools may underestimate how much balancing power is required before the next control interval. Therefore, the design of the consistency term is to penalize unsupported oscillations more severely than a physically reasonable slope.

### Adaptive Consistency Objective

The last one is adaptive weighting instead of a fixed smoothing factor. Periodically, when clear and gradually changing, a larger temporal penalty is applied; otherwise, it is not. During a rapid increase in irradiance, the model is given a lower smoothness weight to keep the actual ramp. It should be noted that this is not a general curve smoother but a consistency-aware forecasting objective.

$$\lambda_t = \lambda_0 exp(-\gamma|r_t|) \quad \text{Eq.(8)}$$

The exponential form reduces excessive smoothing during genuine ramps and keeps a stronger consistency signal during stable intervals.

$$L_{TC} = L_{acc} + \lambda_t L_{ramp} + \beta L_{curv} \quad \text{Eq.(9)}$$

The exponential form reduces excessive smoothing during genuine ramps and keeps a stronger consistency signal during stable intervals.

Adaptive Design can be used to improve station pooling. A fixed penalty selected for a stable rooftop plant may be too large for a plant exposed to rapid cloud shadows, and a penalty chosen for a volatile plant may be too small for a smooth clear-sky station. Connect the weight to the recent ramp evidence, and then a local adjustment can be obtained without needing a separate temporal-consistency coefficient for each station. Therefore, the training period will be relatively short, and less site-specific optimization will be required.

## TC-BiLSTM Forecasting Architecture

### Forecasting Workflow

The TC-BiLSTM architecture is organized as a data-quality layer, a temporal encoder, a horizon-aware decoder, and a consistency-enhanced learning objective. Figure 1 is reserved for the complete forecasting workflow, including station data ingestion, quality control, feature construction, model training, and rolling deployment. The workflow begins with raw SCADA and meteorological streams, removes night-time inactive samples from metric computation, interpolates short missing segments, and flags longer gaps for exclusion. Each station keeps its own capacity and location descriptors, while the recurrent model is trained on normalized sequences pooled across stations.

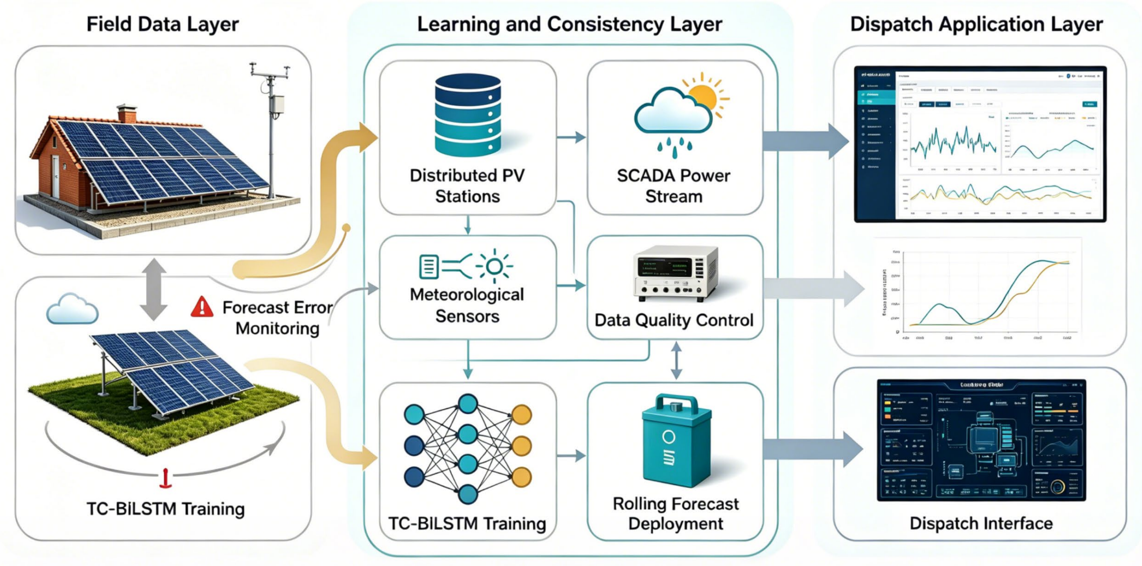


**Figure 1.** End-to-end workflow of temporal-consistency-enhanced distributed PV power forecasting.

The encoder uses two LSTM chains running in opposite directions across the input window. The forward chain captures chronological evolution from older observations to the current state, and the backward chain allows each hidden representation to account for the full observed window. Their hidden states are concatenated and passed to a compact projection layer. This design is deliberately moderate in size because distributed PV stations often require frequent retraining as sensors are added, replaced, or recalibrated.

Feature construction is intentionally conservative. The model uses sine and cosine encodings for time of day and day of year, capacity-normalized lagged power, irradiance derivatives, recent clear-sky-index statistics, and binary missingness flags. It does not require sky images, satellite cloud vectors, or proprietary weather products. Those data sources can be valuable, but the aim here is a forecasting method that can be deployed with the measurement channels commonly available in distributed PV station management systems.

### Bidirectional Encoder and Horizon-Aware Decoder

$$h_{f,t} = LSTM_f\left(x_t, h_{f,t-1}\right) \qquad \text{Eq.(10)}$$

The decoder uses this representation to assign a distinct context to each forecasting horizon while retaining the latest ramp state.

$$h_{b,t} = LSTM_b\left(x_t, h_{b,t+1}\right) \qquad \text{Eq.(11)}$$

The decoder uses this representation to assign a distinct context to each forecasting horizon while retaining the latest ramp state.

$$z_t = W_z\left[h_{f,t}; h_{b,t}\right] + b_z \qquad \text{Eq.(12)}$$

The decoder uses this representation to assign a distinct context to each forecasting horizon while retaining the latest ramp state.

$$a_h = softmax\left(q_h^T tanh(W_a z_t + U_a e_h)\right) \qquad \text{Eq.(13)}$$

The decoder uses this representation to assign a distinct context to each forecasting horizon while retaining the latest ramp state.

A horizon embedding is used because the uncertainty and temporal structure of a 15-minute forecast are different from those of a 4-hour forecast. Therefore, the decoder does not copy the same latent vector for all horizons. It builds a horizon-specific context vector, adds the latest ramp descriptor to it, and then maps the result to a bounded-capacity-normalized output. Figure 2 is used to show the structure of the TC-BiLSTM model, including the bidirectional encoder, horizon attention, temporal-consistency module and output sequence.

https://doi.org/10.64972/jaat.2024v2.316p26e:361-373

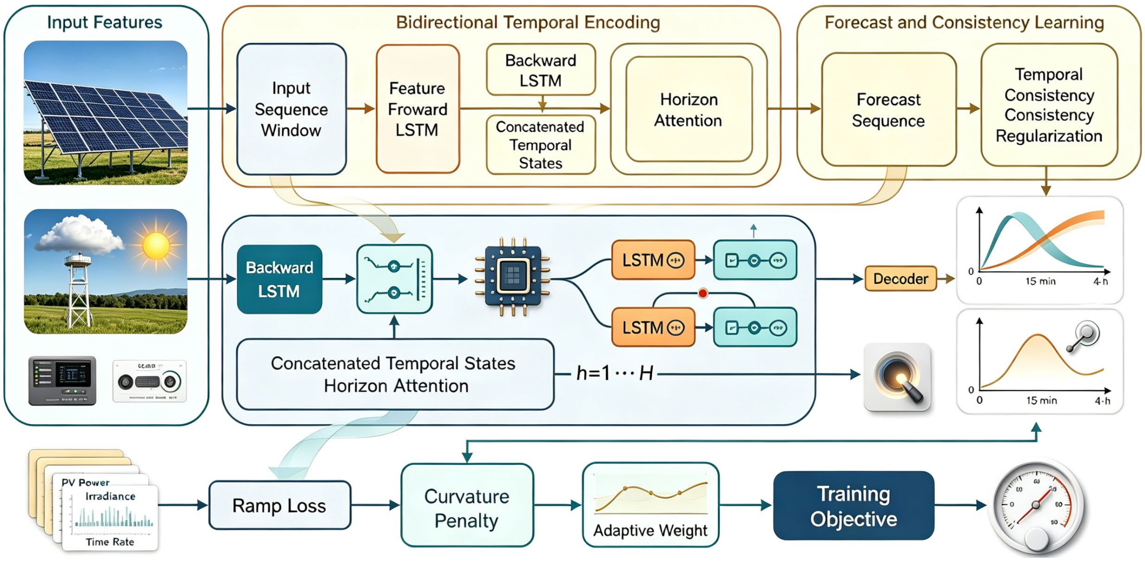


**Figure 2.** TC-BiLSTM architecture with bidirectional temporal encoding, horizon-aware decoding, and consistency regularization.

Training is performed with rolling-origin validation. The chronological split avoids leakage from future weather regimes into the training period. Mini-batches are station-balanced so that stations with dense records do not suppress smaller sites. The model is optimized with AdamW, early stopping on validation RMSE, and gradient clipping. Hyperparameters include input length $L = 32$, forecast horizon $H = 16$, hidden size 96, dropout 0.18, base temporal weight 0.25, curvature weight 0.06, and batch size 128.

The training protocol also has capacity-stratified sampling. Small stations are more likely to have sensor quantization and shading; generally, the aggregate output of larger stations is smoother. If the batch is only selected based on the number of records, the network may learn a representation skewed towards frequently sampled and large-scale plants. Stratified sampling allocates small, medium and large stations in each mini-batch to improve transfer performance for newly added plants with short histories.

**Training and Deployment Procedure**

$$\theta^* = argmin_\theta(1/N) \sum_{(X,y)\in D_{train}} L_{TC}(\theta; X, y) \qquad \text{Eq.(14)}$$

These metrics are computed on active daytime samples and reported in capacity-normalized units for station-level comparability.

$$RMSE = \sqrt{(1/N) \sum_{i=1}^{N} (y_i - \hat{y}_i)^2} \qquad \text{Eq.(15)}$$

These metrics are computed on active daytime samples and reported in capacity-normalized units for station-level comparability.

Deploy the most recently validated feature window at each station during the deployment process and refresh the forecast every 15 minutes. If a meteorological variable is unavailable, the model will use the station's imputation profile and add a missingness flag to the feature vector. The consistency term is employed in training, and inference is a single forward pass. Therefore, this way is still suitable for feeder-level scheduling systems that need to update the entire batch of station forecasts in seconds.

## Experimental Data and Performance Analysis

### Data Description and Experimental Setup

The experiments use measurements from 38 distributed PV stations connected to three medium-voltage feeders. The total installed capacity is 42.6 MW, with station ratings ranging from 0.42 MW to 2.35 MW. The sampling interval is 15 min, producing 96 records per day. After removing night-time inactive intervals from metric computation, the dataset contains 1.18 million valid daytime samples. The first 12 months are used for training, the next 3 months for validation, and the final 3 months for testing. Weather labels are assigned from irradiance variability and daily clearness index: 41.3% clear, 29.6% partly cloudy, 20.4% overcast, and 8.7% rainy. The data profile is referenced once in Figure 3: Figure 3(a) shows that clear-day normalized power peaks near 0.91 while rainy days rarely exceed 0.34; Figure 3(b) shows the densest irradiance-power response between 550 and 820 W/m2; after the data-cleaning rule used in recent PV forecasting experiments [15], Figure 3(c) shows that the 95th-percentile absolute ramp rate reaches 0.184 per 15 min in the eastern station group, compared with 0.137 in the western group.

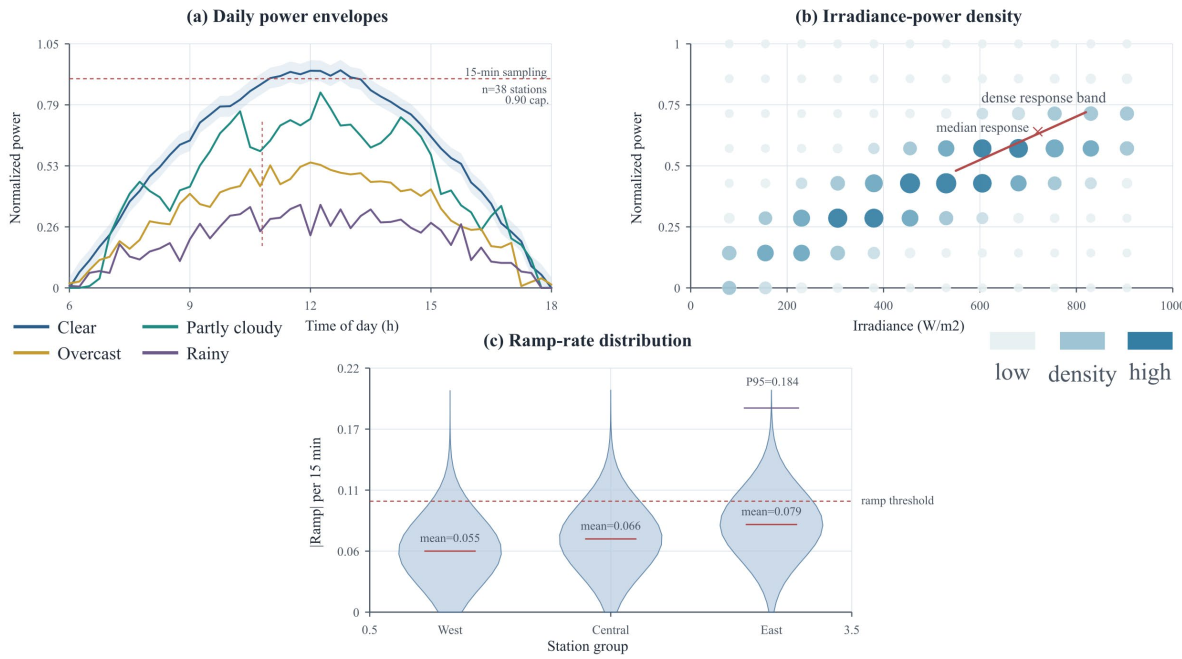


**Figure 3.** Multi-station data characteristics: (a) normalized daily power envelopes under four weather regimes; (b) irradiance-power response density; (c) ramp-rate distribution by station group.

Six baseline models are implemented with the same training, validation, and test split: persistence, SVR, random forest, CNN-LSTM, LSTM, and standard BiLSTM. All neural models use the same input variables, and their hyperparameters are tuned on the validation set following short-term PV feature-selection practice [16]. The proposed TC-BiLSTM achieves the lowest overall error, with RMSE of 6.58% capacity, MAE of 3.91% capacity, and MAPE of 8.46% on active daytime samples. Standard BiLSTM reaches 7.14% RMSE and 4.28% MAE, while CNN-LSTM reaches 7.63% RMSE and 4.56% MAE. The persistence baseline remains competitive at the first 15-min horizon but deteriorates rapidly after 1 h, reaching 12.81% RMSE at the 4-h horizon.

The Error Distribution is not uniform by station capacity. Stations with capacity less than 0.8MW have a record RMSE of 7.12% in the proposed method and 7.89% in the standard BiLSTM. For stations with a capacity of 0.8 MW - 1.5 MW, the reduction is 7.03% - 6.44%. Stations over 1.5 MW have reduced the rate by 0.46%, from 6.58% to 6.12%. Therefore, the improvement is relatively smaller at the small station, as measurement noise and local shading cause a more uneven forecast curve. Therefore, a consistency term can be used to reduce local jitter without adding an extra denoising model.

Verify the stability of the indicators during the three-month testing period to prevent results from being influenced by unusually favorable periods. TC-BiLSTM achieved an RMSE of 6.44% and an MAE of 3.82% in the

https://doi.org/10.64972/jaat.2024v2.316p26e:361-373

first test month; in the second month, it was 6.71% RMSE and 4.03% MAE; and in the third month, it reached 6.60% RMSE and 3.88% MAE. The standard deviation of monthly RMSE is 0.14 percentage points, and the corresponding standard deviation for the standard BiLSTM is 0.27 percentage points. Lower monthly fluctuations are convenient for engineering deployment because model acceptance typically requires long-term behavioral performance rather than a single comprehensive test result.

**Forecasting Accuracy and Temporal Consistency**

The horizon-wise comparison is referenced once in Figure 4: Figure 4(a) shows that TC-BiLSTM keeps RMSE below 5.1% for the first hour and below 8.2% at 4 h, whereas standard BiLSTM rises to 9.0% at 4 h; in line with the emphasis on model-level PV forecast comparisons [17], Figure 4(b) shows that partly cloudy MAE is reduced from 6.42% to 4.97%; Figure 4(c) shows the median station nRMSE falling from 7.02% to 6.31%; Figure 4(d) reports a persistence-relative skill score of 31.8% at 2 h and 36.5% at 4 h. These gains indicate that the temporal-consistency term helps most when the forecast horizon extends beyond the range where persistence alone is informative.

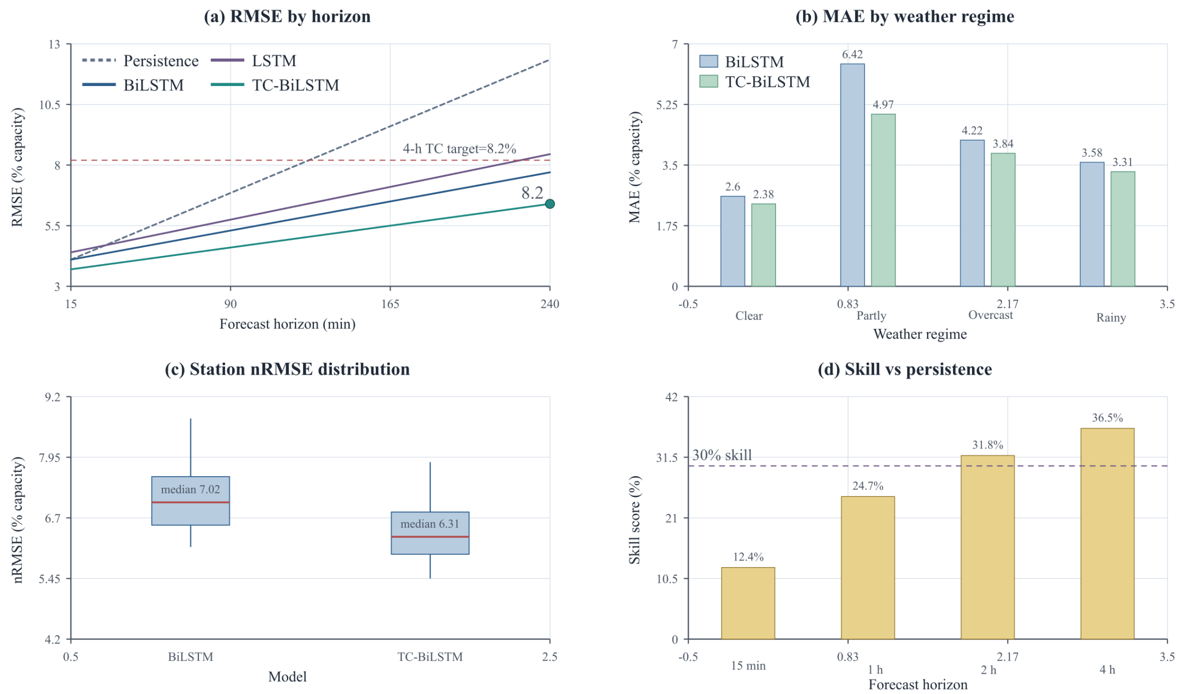


**Figure 4.** Forecasting accuracy comparison: (a) RMSE by horizon; (b) MAE by weather regime; (c) nRMSE distribution across stations; (d) skill score relative to persistence.

Weather-regime analysis gives a more practical view of the improvement. On clear days, the RMSE reduction relative to standard BiLSTM is modest, from 4.21% to 3.92%, because the daily power curve is already smooth and strongly determined by solar geometry. On overcast days, the reduction is from 5.88% to 5.41%. The largest improvement occurs on partly cloudy days, where intermittent cloud movement creates steep ramps and short reversals. The short-horizon setting here is more sensitive to ramp timing than day-ahead nonlinear PV prediction [18]. In that regime, TC-BiLSTM reduces RMSE from 9.78% to 7.83% and decreases the mean absolute ramp error from 0.071 to 0.055 capacity per 15 min.

The partly cloudy subset is examined further because it is the main stress case for short-term PV forecasting. During the 312 test days classified as partly cloudy at one or more stations, 18.6% of active intervals contain a ramp above 0.10 capacity per 15 min. Standard BiLSTM tends to lag the first ramp step and then compensates with a delayed overshoot. TC-BiLSTM reduces this behavior: the median ramp-timing error decreases from two sampling intervals to one, and the 90th-percentile overshoot falls from 0.126 to 0.088 capacity. The improvement is visible even when the final point error is similar because the predicted curve arrives at the event with a more realistic slope.

https://doi.org/10.64972/jaat.2024v2.316p26e:361-373

Directly assess temporal consistency, do not infer from point errors. Set the threshold for slope events to an absolute capacity normalized change of 0.10 within 15 minutes. Based on the above definition, TC-BiLSTM has a ramp precision of 0.836, a recall of 0.811 and an F1-score of 0.823. Standard BiLSTM: Precision = 0.742, Recall = 0.670, F1-score = 0.704. Smoothness analysis is referenced once in Figure 5: Figure 5(a) shows that the average adjacent-horizon ramp error decreases from 0.046 to 0.034; using the hourly photovoltaic forecasting perspective [19], Figure 5(b) shows that the curvature penalty reduced the isolated one-step peak by 28.9%; Figure 5(c) shows that the best trade-off is at a temporal weight of 0.20 to 0.30, and accuracy improves without flattening real ramps.

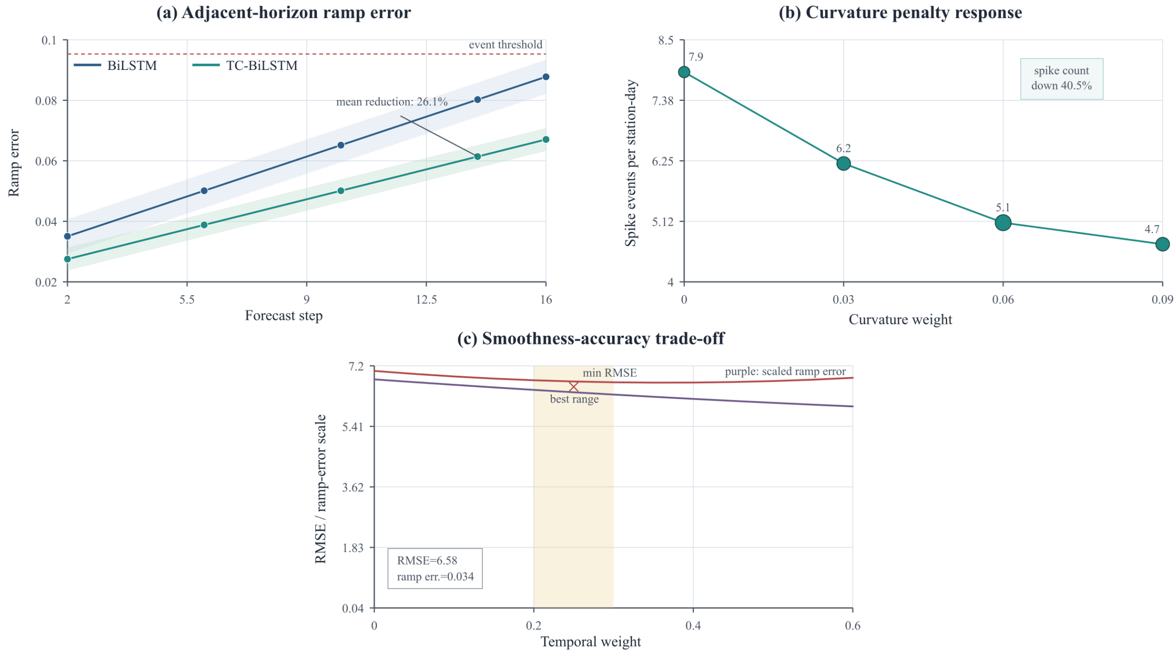


**Figure 5.** Temporal-consistency behavior: (a) adjacent-horizon ramp error; (b) curvature penalty response; (c) forecast smoothness versus accuracy trade-off.

The benefit also appears in feeder aggregation. When station forecasts are summed by feeder, random station-level errors partly cancel, but timing errors in cloud-driven ramps remain visible. TC-BiLSTM reduces feeder-level RMSE from 5.46% to 4.82% and lowers the maximum 1-h energy deviation from 1.37 MWh to 0.98 MWh. This energy-oriented metric is important because dispatchers often allocate flexibility over blocks of several intervals rather than reacting to each individual 15-min point. The result shows that temporal consistency improves both the instantaneous forecast curve and the accumulated energy estimate used by practical scheduling tools.

Ablation experiments separate the contribution of the main components. Removing the ramp consistency term increases RMSE from 6.58% to 6.94%. Removing the curvature term raises the spike count from 4.7 to 7.9 events per station-day. Replacing the bidirectional encoder with a one-directional LSTM increases RMSE to 7.25%, and removing horizon attention increases RMSE to 6.86%. The ablation and sensitivity results are referenced once in Figure 6: Figure 6(a) reports the component-wise RMSE changes; Figure 6(b) shows that the 32-step input window performs better than 16 and 64 steps; Figure 6(c) shows that temporal weight above 0.45 begins to over-smooth fast ramps while hidden size 96 remains the best accuracy-complexity point, which is consistent with hybrid hour-ahead PV studies [20].

The input-window result deserves attention because longer history is not always better. A 16-step window misses the gradual pre-ramp attenuation that often appears before cloud-edge events, leading to 6.91% RMSE. A 32-step window captures that transition while keeping the feature sequence compact. Input history must be tuned to the operational horizon in LSTM-based PV forecasting [21]. Here, a 64-step window slightly degrades validation error to 6.76%, mainly because the older samples add regime information that is weakly related to the next 4 h. This finding supports an engineering preference for moderate input windows in distributed PV stations where real-time retraining and memory footprint matter.

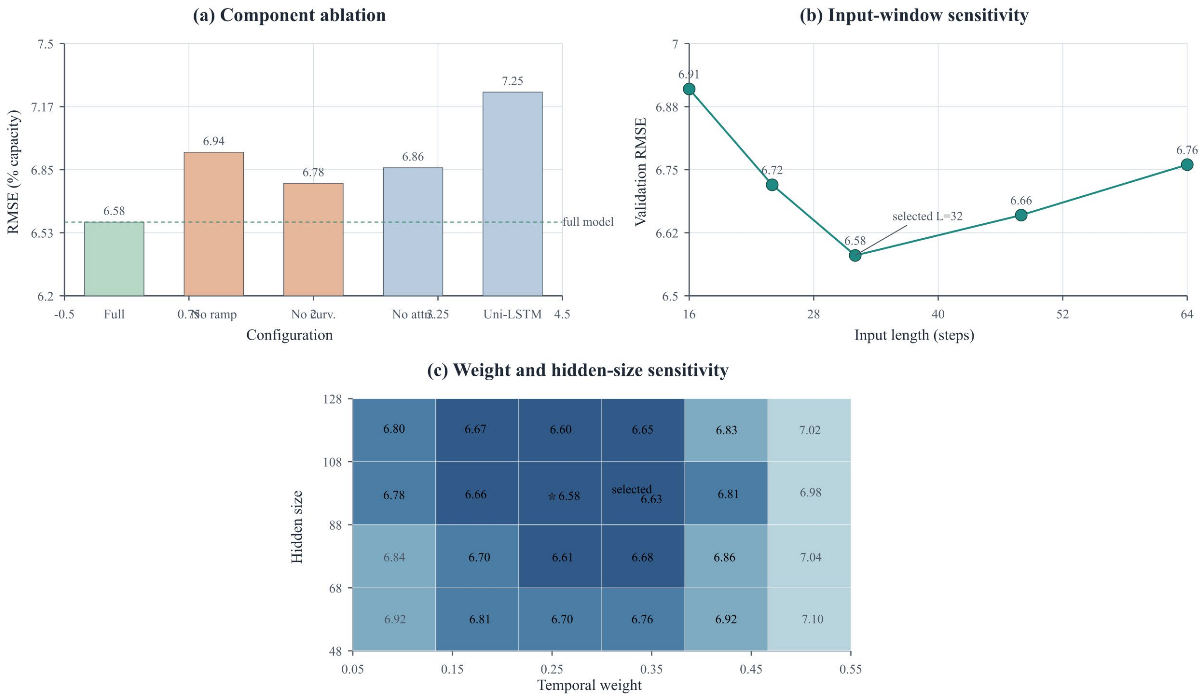

**Figure 6.** Ablation and sensitivity analysis: (a) component ablation; (b) input-window sensitivity; (c) temporal-weight and hidden-size sensitivity.

## Ablation, Sensitivity, and Operational Cases

Operational cases are analyzed using representative days from the test set. The case trajectories are referenced once in Figure 7: Figure 7(a) shows clear, partly cloudy, rainy, and feeder-aggregated trajectories, where peak clear-day error remains within 2.8% capacity and the partly cloudy event includes a 0.31 capacity drop over 45 min; Figure 7(b) shows that 91.4% of TC-BiLSTM residuals fall inside the +/-0.08 dispatch band, compared with 84.6% for standard BiLSTM. This case-based evaluation follows the practical emphasis of multiscale very-short-term PV forecasting [22].

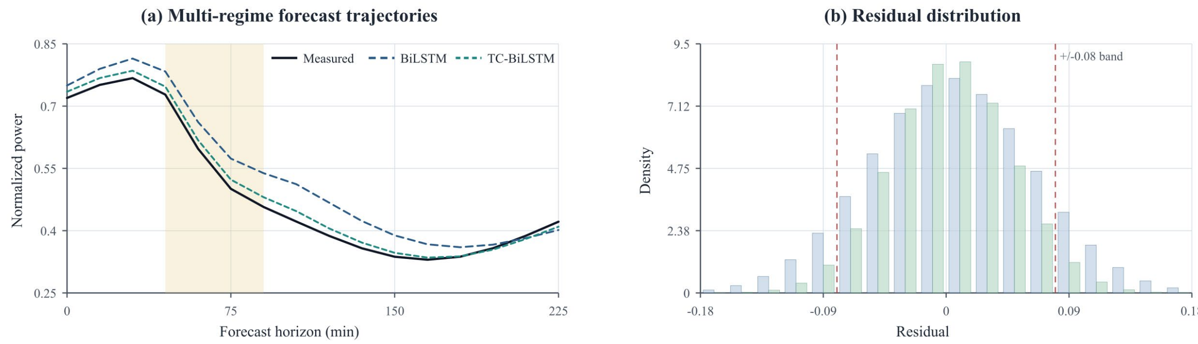

**Figure 7.** Operational forecasting cases: (a) multi-regime forecast trajectories; (b) residual distribution with dispatch bands.

Therefore, it is also more stable in terms of forecasting according to time. When a real ramp exists, the adaptive weight will be reduced based on the size of the recent ramp, and thus the decoder can follow the event. Increase the penalty strength and eliminate artificial oscillations during a stable period. The multi-model photovoltaic forecasting work [23] provides motivation for the division between stability and responsiveness; here, it is explained why the model improves both MAE and ramp-event F1-score, a combination that is difficult to achieve with simple moving-average post-processing.

Directly compare it with moving-average post-processing to verify the above. A three-point moving average is added to the standard BiLSTM, and the average curvature index is reduced by 19.4%; however, due to the delay in real-time cloud-edge changes, the RMSE rises to 7.31% from 7.14%. The same filter applied to TC-BiLSTM only reduces the curvature slightly and slightly increases MAE. The method proposed thus far has learned where smoothness is needed during training, so a blind post-processing filter adds little benefit and may alter ramp timing.

Efficient real-time scheduling. On the workstation-level CPU, a rolling update for all 38 sites takes 0.82 seconds and includes feature normalization and missing flag construction. GPU inference reduces the same update time to 0.11s. The trained model has 0.74 million parameters and is smaller than the tested CNN-LSTM configuration with 1.18 million parameters. Lightweight deep-learning PV predictors have shown support for a short update cycle [24]. Distribution Operators require high-speed networks that can also guarantee relatively low latency before the start of the next dispatch cycle.

Several error patterns can still be observed. First, if the operational flag is delayed, sudden inverter clipping after maintenance or reduction may be mistakenly identified as weather-related behavior. Second, snow cover and soiling events are underrepresented in the dataset and cause a continuous negative bias. Third, stations with unstable irradiance sensors are still included in the shared model, but their station-level error is relatively high compared with the rest of the feeder. Data uncertainty cannot be addressed by the predictor alone, and therefore, data governance and event labelling still need to be employed with temporal models [25].

The station-transfer test provides a final robustness check. Five stations are excluded from training and used only during testing with shared normalization statistics and station metadata. TC-BiLSTM reaches 6.95% RMSE on these unseen stations, compared with 7.41% for standard BiLSTM and 8.02% for CNN-LSTM. The small degradation relative to the ordinary test set suggests that the consistency objective captures a transferable property of PV trajectories rather than memorizing station identity. The result is relevant for distribution companies that continually connect new rooftop clusters and cannot wait for a full year of local history before using a model in scheduling support.

## Conclusion

This paper proposes TC-BiLSTM, a method for predicting distributed photovoltaic stations that enhances temporal consistency. This method combines bidirectional recurrent encoding, horizon-aware decoding, and adaptive consistency objectives to adjust the slope and curvature of predictions while preserving the true cloud-driven transitions. This study incorporates temporal consistency into the training objectives, enabling the model to improve point accuracy and generate prediction trajectories that are easier to use in distribution system scheduling.

This method still has limitations. It relies on reliable time synchronization between site power, irradiance, and weather measurements, and its adaptive consistency weights are derived from recent fluctuation behavior rather than from an explicit cloud motion model. Rare operating conditions, such as power outages, snow accumulation, severe dirt, and sensor replacements, may require additional labels or separate calibration modules before deployment in areas where these events frequently occur.

Future research will extend this method to consider time consistency under uncertainty, graph-based site coupling, and online adaptation under changing site combinations. Another useful direction is to connect the prediction objectives with downstream feeder optimization, so as to learn prediction smoothness, slope fidelity, and scheduling costs within a single operational framework.

**Author Contributions**

Rasha Al-Qarni contributes to conceptualization, methodology, software, validation, analysis, investigation, data collection, draft preparation, manuscript editing, visualization, supervision. Zuhaira Al-Buainain contributes to data collection, draft preparation, manuscript editing. Obeida Al-Zabeidi contributes to data collection. All authors have read and agreed with the manuscript before its submission and publication.

**Funding**

This research received no specific financial support from any funding agency.

**Institutional Review Board Statement**

Not applicable.

## References

[1] Luo, X., Zhang, D., & Zhu, X. (2021). Deep learning based forecasting of photovoltaic power generation by incorporating domain knowledge. Energy, 225, 120240. https://doi.org/10.1016/j.energy.2021.120240

[2] Guo, X., Mo, Y., & Yan, K. (2022). Short-Term Photovoltaic Power Forecasting Based on Historical Information and Deep Learning Methods. Sensors, 22(24), 9630. https://doi.org/10.3390/s22249630

[3] Mellit, A., Pavan, A. M., & Lughi, V. (2021). Deep learning neural networks for short-term photovoltaic power forecasting. Renewable Energy, 172, 276-288. https://doi.org/10.1016/j.renene.2021.02.166

[4] Li, G., Xie, S., Wang, B., Xin, J., Li, Y., & Du, S. (2020). Photovoltaic Power Forecasting With a Hybrid Deep Learning Approach. IEEE Access, 8, 175871-175880. https://doi.org/10.1109/access.2020.3025860

[5] Tang, Y., Yang, K., Zhang, S., & Zhang, Z. (2022). Photovoltaic power forecasting: A hybrid deep learning model incorporating transfer learning strategy. Renewable and Sustainable Energy Reviews, 162, 112473. https://doi.org/10.1016/j.rser.2022.112473

[6] Luo, X., & Zhang, D. (2022). An adaptive deep learning framework for day-ahead forecasting of photovoltaic power generation. Sustainable Energy Technologies and Assessments, 52, 102326. https://doi.org/10.1016/j.seta.2022.102326

[7] Dairi, A., Harrou, F., Sun, Y., & Khadraoui, S. (2020). Short-Term Forecasting of Photovoltaic Solar Power Production Using Variational Auto-Encoder Driven Deep Learning Approach. Applied Sciences, 10(23), 8400. https://doi.org/10.3390/app10238400

[8] Rajagukguk, R. A., Ramadhan, R. A. A., & Lee, H. J. (2020). A Review on Deep Learning Models for Forecasting Time Series Data of Solar Irradiance and Photovoltaic Power. Energies, 13(24), 6623. https://doi.org/10.3390/en13246623

[9] Massaoudi, M., Chihi, I., Abu-Rub, H., Refaat, S. S., & Oueslati, F. S. (2021). Convergence of Photovoltaic Power Forecasting and Deep Learning: State-of-Art Review. IEEE Access, 9, 136593-136615. https://doi.org/10.1109/access.2021.3117004

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

[11] Balraj, G., Victoire, A. A., S., J., & Victoire, A. (2022). Variational mode decomposition combined fuzzy-Twin support vector machine model with deep learning for solar photovoltaic power forecasting. PLOS ONE, 17(9), e0273632. https://doi.org/10.1371/journal.pone.0273632

[12] Huang, Y., Zhou, M., & Yang, X. (2022). Ultra-short-term photovoltaic power forecasting of multifeature based on hybrid deep learning. International Journal of Energy Research, 46(2), 1370-1386. https://doi.org/10.1002/er.7254

[13] Zang, H., Cheng, L., Ding, T., Cheung, K. W., Wei, Z., & Sun, G. (2020). Day-ahead photovoltaic power forecasting approach based on deep convolutional neural networks and meta learning. International Journal of Electrical Power & Energy Systems, 118, 105790. https://doi.org/10.1016/j.ijepes.2019.105790

[14] Rodríguez, F., Azcárate, I., Vadillo, J., & Galarza, A. (2022). Forecasting intra-hour solar photovoltaic energy by assembling wavelet based time-frequency analysis with deep learning neural networks. International Journal of Electrical Power & Energy Systems, 137, 107777. https://doi.org/10.1016/j.ijepes.2021.107777

[15] Xu, D., Shao, H., Deng, X., & Wang, X. (2022). The Hidden-Layers Topology Analysis of Deep Learning Models in Survey for Forecasting and Generation of the Wind Power and Photovoltaic Energy. Computer Modeling in Engineering & Sciences, 131(2), 567-597. https://doi.org/10.32604/cmes.2022.019245

[16] Wu, K., Peng, X., Li, Z., Cui, W., Yuan, H., Lai, C. S., & Lai, L. L. (2022). A Short-Term Photovoltaic Power Forecasting Method Combining a Deep Learning Model with Trend Feature Extraction and Feature Selection. Energies, 15(15), 5410. https://doi.org/10.3390/en15155410

[17] Cordeiro-Costas, M., Villanueva, D., Eguía-Oller, P., & Granada-Álvarez, E. (2022). Machine Learning and Deep Learning Models Applied to Photovoltaic Production Forecasting. Applied Sciences, 12(17), 8769. https://doi.org/10.3390/app12178769

[18] Wang, M., Wang, P., & Zhang, T. (2022). Evidential Extreme Learning Machine Algorithm-Based Day-Ahead Photovoltaic Power Forecasting. Energies, 15(11), 3882. https://doi.org/10.3390/en15113882

[19] Ren, X., Zhang, F., Zhu, H., & Liu, Y. (2022). Quad-kernel deep convolutional neural network for intra-hour photovoltaic power forecasting. Applied Energy, 323, 119682. https://doi.org/10.1016/j.apenergy.2022.119682

[20] Massaoudi, M., Abu-Rub, H., Refaat, S. S., Trabelsi, M., Chihi, I., & Oueslati, F. S. (2021). Enhanced Deep Belief Network Based on Ensemble Learning and Tree-Structured of Parzen Estimators: An Optimal Photovoltaic Power Forecasting Method. IEEE Access, 9, 150330-150344. https://doi.org/10.1109/access.2021.3125895

[21] Akhter, M. N., Mekhilef, S., Mokhlis, H., Ali, R., Usama, M., Muhammad, M. A., & Khairuddin, A. S. M. (2022). A hybrid deep learning method for an hour ahead power output forecasting of three different photovoltaic systems. Applied Energy, 307, 118185. https://doi.org/10.1016/j.apenergy.2021.118185

[22] Kim, D., Kwon, D., Park, L., Kim, J., & Cho, S. (2020). Multiscale LSTM-Based Deep Learning for Very-Short-Term Photovoltaic Power Generation Forecasting in Smart City Energy Management. IEEE Systems Journal, 15(1), 346-354. https://doi.org/10.1109/jsyst.2020.3007184

[23] Polo, A. (2022). A TWO-STEP LEARNING-BY-EXAMPLES METHOD FOR PHOTOVOLTAIC POWER FORECASTING. Progress In Electromagnetics Research C, 125, 35-49. https://doi.org/10.2528/pierc22061003

[24] Elsaraiti, M., & Merabet, A. (2022). Solar Power Forecasting Using Deep Learning Techniques. IEEE Access, 10, 31692-31698. https://doi.org/10.1109/access.2022.3160484

[25] Gao, W., & Chen, Q. (2020). An Bayesian Learning and Nonlinear Regression Model for Photovoltaic Power Output Forecasting. Applied Mathematics and Nonlinear Sciences, 5(2), 531-542. https://doi.org/10.2478/amns.2020.2.00032