sustainability

MDPI

Article

Lightweight Multimode Day-Ahead PV Power Forecasting for Intelligent Control Terminals Using CURE Clustering and Self-Updating Batch-Lasso

Ting Yang [1,*], Butian Chen [1], Yuying Wang [2], Qi Cheng [1] and Danhong Lu [1]

[1] School of Electric Power Engineering & School of Shen Guorong, Nanjing Institute of Technology, Nanjing 211167, China
[2] Xuzhou Jiawang District Power Supply Branch, National Grid Jiangsu Electric Power Co., Ltd., Xuzhou 221000, China
* Correspondence: tyang@njit.edu.cn

Abstract

Lightweight day-ahead photovoltaic (PV) forecasting models encounter a significant technical challenge: under resource-constrained deployment conditions, it is difficult to simultaneously address weather-regime heterogeneity, maintain model interpretability, and preserve adaptability as operating conditions evolve. To address this issue, we propose a multimodal short-term photovoltaic (PV) forecasting method that integrates weather-mode partitioning using the Clustering Using Representatives (CURE) algorithm with a self-updating Batch-Lasso model. First, the meteorological-PV dataset is partitioned along two dimensions by combining seasonal grouping with CURE clustering within each season, producing representative weather modes and enhancing the fidelity of weather pattern classification. Second, to extract informative predictors from high-dimensional meteorological inputs while maintaining interpretability, we formulate per-mode Lasso regression and adopt the Fast Iterative Shrinkage-Thresholding Algorithm (FISTA) to efficiently solve for the sparse regression coefficients. Third, we introduce a batch-based self-update and correction mechanism with rollback verification, enabling the mode-specific models to be refreshed as new historical data become available while preventing performance degradation. Compared with representative machine learning baselines, the proposed method maintains competitive accuracy with substantially lower computational and storage overhead, enabling high-frequency and energy-efficient inference on resource-constrained terminals, thereby reducing operational burdens and computational energy costs and better meeting the deployment needs of sustainable energy systems under heterogeneous weather conditions.

Keywords: lightweight multimode day-ahead PV power forecasting; CURE; self-updating Batch-Lasso; FISTA; intelligent control terminals; energy-efficient inference; sustainable energy systems

Check for updates

Academic Editor: Carlos Vargas-Salgado

Received: 13 February 2026
Revised: 19 March 2026
Accepted: 26 March 2026
Published: 29 March 2026

1. Introduction

1.1. Background and Motivation

China's PV installed capacity has experienced rapid growth in recent years. Annual new installations of distributed PV surged from 15.5 GW in 2020 to 153.25 GW in 2025. By the end of 2025, the cumulative PV capacity in China is projected to reach 1200 GW, with distributed PV capacity accounting for 530 GW [1]. Consequently, a substantial number of distributed PV systems are connected at the extremities of the distribution network, resulting in a pattern characterized by disorder and widespread dispersion. Furthermore, PV power generation is intermittent, fluctuating, and highly unpredictable [2–4]. To enhance the capacity for PV consumption and achieve intelligent dispatch and optimal energy utilization, accurate day-ahead short-term forecasting of distributed PV power is essential. Presently, numerous intelligent control terminals (ICTs) situated at the edge of the power grid are capable of managing flexible resources, including sources, loads, and storage. Utilizing the edge computing capabilities of these systems to forecast short-term PV power in real-time is crucial for enabling intelligent power control [5–7].

1.2. Literature Review

Day-ahead short-term PV power generation is significantly influenced by meteorological conditions across various weather modes. Exploring the quantitative relationships between these meteorological factors and PV power is crucial for enhancing forecasting accuracy. Three primary technological approaches are commonly employed: (i) improving the selection of characteristic meteorological variables through correlation analysis and principal component analysis [8–10]; (ii) incorporating additional meteorological factors [11,12]; and (iii) utilizing clustering analysis to aggregate data from days similar to the forecast day. Previous research has demonstrated a strong correlation between meteorological factors such as solar irradiance, temperature, humidity, wind speed, wind direction, and atmospheric pressure with PV power generation [13–16]. Various algorithms, including K-means clustering [17], Gaussian mixture model clustering [18,19], fuzzy C-means clustering [20], and dynamic self-organizing map clustering [21], have been utilized to analyze PV power datasets. However, current clustering methods often categorize weather types broadly and lack detailed characteristics.

In addition to selecting the characteristics of meteorological factors, choosing the appropriate construction method for the PV forecasting model is essential. Different construction methods directly impact forecasting accuracy and generalization ability, while the complexity of the model also affects its deployment and practical application. Currently, forecasting models are primarily categorized into two types: regression analysis and neural networks [22–24]. PV power data typically follows a time series pattern. Recent research has focused on neural network models tailored for time series analysis, including recurrent neural networks (RNNs) [25], long short-term memory networks (LSTMs) [26–28], transformers, generative adversarial networks [29], and mixed models [30,31], among others.

Neural network models are characterized by their complexity and limited interpretability. Training these models typically demands substantial computational resources, and hyperparameter settings are often established based on empirical experience, which can limit the model's applicability [32]. Furthermore, even after a model has been trained and tested, it must undergo an intricate training and testing process during subsequent applications. While model accuracy is a critical concern, it is equally important to evaluate its suitability for deployment and its capacity for self-updating [33–35]. Given that edge computing resources in ICTs are limited, regression models are more conducive to edge deployment. Compared to neural networks, regression models offer lower complexity and stronger interpretability. Although early research focused on regression analysis models, the growing interest in the nonlinear fitting capabilities of neural networks has led to reduced exploration of regression models in later stages [36]. Classic linear regression techniques include simple linear regression, ridge regression, lasso regression, elastic net regression, Bayesian ridge regression, minimum angle regression, and partial least squares regression, among others. Several PV forecasting models utilizing linear regression are discussed in the literature [37–39]. The research methodology involves extracting independent variables related to meteorological factors using principal component analysis or correlation analysis. Subsequently, models are fitted using techniques such as linear regression, support vector machines, and random forests.

In summary, existing multimode PV forecasting studies primarily enhance predictive homogeneity through methods such as weather clustering or similar-day partitioning. However, a significant challenge remains in how forecasting models can adapt to evolving operating conditions over time. Recent studies have underscored the necessity of online-updated forecasting in response to changing data distributions [40]. Concurrently, regression-based PV forecasting continues to be appealing in practical applications due to its interpretability, low computational burden, and feasibility in resource-constrained environments [41,42]. Therefore, while existing studies have individually examined multimodal prediction and adaptive prediction, there is a notable lack of research that concurrently integrates weather pattern classification, sparse regression modeling, lightweight deployment, and self-updating capabilities into a cohesive framework. To further clarify the novelty and positioning of this study, Appendix A (Table A1) provides a comparative summary of representative prior studies and the present work.

1.3. Organization

The organization of this paper is structured as follows: Section 2 describes the dual-dimension weather mode partitioning method, which is based on seasonal grouping and CURE clustering. Section 3 presents the per-mode Lasso/Batch-Lasso forecasting formulation, including FISTA-based coefficient estimation and the proposed self-updating mechanism. Section 4 summarizes the implementation details and evaluation protocol. Finally, Section 5 reports the case studies and comparative experiments conducted to validate the effectiveness and practicality of the proposed framework.

2. Weather Multimode Division Based on CURE

PV power generation is primarily influenced by meteorological factors, particularly solar irradiance and temperature. The Earth's rotation induces seasonal variations in these factors, allowing for the categorization of PV power into first-level modes based on seasonal changes.

In addition to seasonal influences, various weather types also directly affect PV power generation. Therefore, clustering is conducted based on the first-level seasonal modes, categorized by different weather types, to establish second-level modes. Given that meteorological datasets describing weather types are characterized by high dimensionality and large volumes of data, traditional clustering methods face several limitations: (i) they are effective only when the data forms spherical clusters; however, meteorological datasets often exhibit clustering that deviates from standard spherical shapes due to their high-dimensional nature; (ii) clustering large datasets tends to converge slowly.

CURE addresses these challenges by performing clustering based on multiple representative points, thereby accommodating diverse cluster shapes. Furthermore, it utilizes a combination of random sampling and data equalization for meteorological datasets, which enhances convergence speed. Consequently, CURE is employed to cluster weather datasets, and the pseudocode for the clustering process is presented in Algorithm 1.

In Algorithm 1, P represents a specific data point, S denotes a seasonal dataset, N refers to the number of equal fraction groups, z can be any data point, N_p is a dataset containing P-points, *Clu-P* signifies a primary cluster, and $l_{\max}$ indicates the maximum number of clusters for the CURE algorithm. The process of identifying and removing outliers using the CURE algorithm is illustrated in Figure 1.

Algorithm 1. Pseudocode for clustering seasonal datasets using CURE

Input: Seasonal Datasets S
Output: Typical Weather Mode Datasets
1 Sample data points from S
2 For each sampled data point P in S do
3   If P is selected in the sampling process then
4   Divide S into N groups
5   Assign P to a group N_p
6   For each data point z in group N_p do
7   Calculate the Euclidean distance between z and P
8   End for
9   Form a primary cluster *Clu-P* with data points close to P
10   Select well-scattered representative points in *Clu-P*
11   Shrink the representative points toward the centroid of *Clu-P*
12   Merge clusters according to the minimum Euclidean distance between representative-point sets
13   Repeat Steps 10–12 until the number of clusters reaches $l_{\max}$
14   Else
15   Assign P to the cluster with the nearest representative-point set
16   End if
17 End for
18 Return typical weather mode datasets

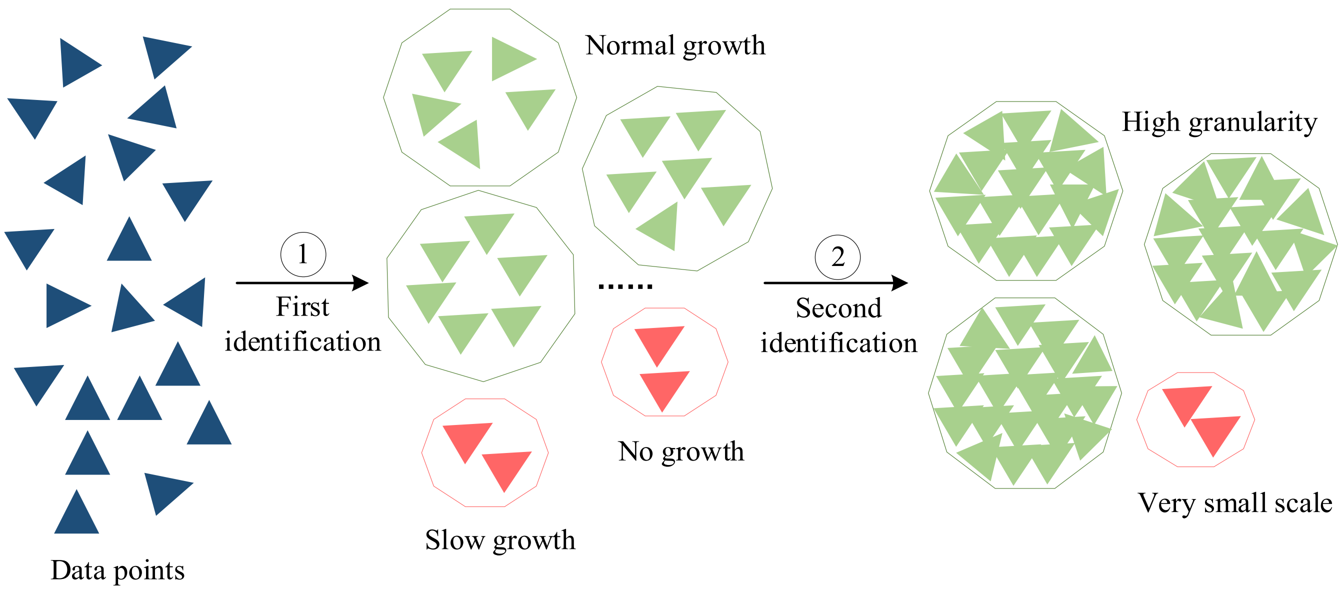


Figure 1. Schematic diagram of CURE identifying and removing outliers.

The identification and removal of outliers by CURE during the clustering process are illustrated in Figure 1. This identification process can be divided into two stages. The first stage occurs when the total number of clusters reaches one-third of the dataset size. At this point, outliers are identified based on the growth rate and size of the clusters; clusters exhibiting very slow growth are classified as outliers and subsequently removed. The second stage occurs when the total number of clusters is reduced to approximately the final clustering count, at which point very small clusters are eliminated. This step is necessary because, in the final stages of clustering, each normal cluster exhibits high granularity, and the proportion of outliers is minimal, making these clusters easily identifiable. The significance of this stage is underscored by the fact that closely distributed outliers may merge into a single cluster, hindering their identification during the first stage.

3. PV Power Forecasting Modeling Based on Batch-Lasso

3.1. Lasso Regression Modeling and Solution

PV power forecasting datasets are characterized by high dimensionality and often face challenges related to variable multicollinearity. In comparison to other regression algorithms, Lasso offers significant advantages in mitigating model overfitting and addressing severe collinearity issues. This method allows for the selection of the most relevant feature variables from high-dimensional meteorological factors, thus facilitating the development of an efficient forecasting model.

Lasso regulates the complexity of the model and the degree of fitting to the target variable by incorporating an L1 regularization term into the loss function. The Lasso model can be expressed as follows:

$$Y = X\beta + \varepsilon \tag{1}$$

$$Q(\beta) = \left|\left|Y - X\beta\right|\right|^2 + \lambda\left|\left|\beta\right|\right|_1 \tag{2}$$

Equation (1) represents the Lasso linear regression equation, while Equation (2) defines $Q(\beta)$ as the loss function. In this context, X denotes the feature matrix, Y represents the observation vector, β indicates the parameter vector, and ε signifies the random error. The parameter λ serves as the penalty factor, which controls the complexity of the Lasso model. As λ increases, a greater penalty is imposed on the model, leading to a simpler model structure.

Due to the presence of the L1 regularization term, the Lasso loss function is non-smooth and not continuously differentiable, which prevents the regression parameters from being obtained using traditional gradient descent methods. To address this issue, we introduce the Fast Iterative Shrinkage-Thresholding Algorithm (FISTA). FISTA effectively solves for the regression parameters by employing a soft thresholding operation through proximal regularization for parameter updates. During the iteration process, FISTA circumvents the challenges associated with inverting large matrices and the non-differentiability of the loss function, effectively generating sparse solutions. Additionally, FISTA incorporates Nesterov acceleration during the gradient descent process, which enhances convergence speed by calculating the gradient of the subsequent step in advance. The iterative steps of FISTA for solving the Lasso parameters are presented in Equations (3)–(5).

$$p_k = \beta_{k-1} + \frac{k-2}{k+1}(\beta_{k-1} - \beta_{k-2}) \tag{3}$$

$$w_k = p_k - t_k X^T (X p_k - y) \tag{4}$$

$$\beta_k = sign(w_k)\max\{|w_k| - t_k\lambda, 0\} \tag{5}$$

where k represents the iteration number, t_k is the step size at the k-th iteration, p_k/w_k is the intermediate variable.

The expression for the $sign(w_k)$ function is defined as

$$sign(w_k) = \begin{cases} 1 & w_k > 0 \\ 0 & w_k = 0 \\ -1 & w_k < 0 \end{cases} \tag{6}$$

In Equation (3)–(5):

(I) The step size t_k is determined using the Barzilai-Borwein (BB) method. Two types of BB step sizes are defined: $(s_k)^T s_k / (s_k)^T b_k$ for even-numbered steps and $(s_k)^T b_k / (b_k)^T b_k$ for odd-numbered steps. The methods for calculating s_k and b_k are illustrated in Equations (7) and (8).

$$s_k = \beta_{k+1} - \beta_k \tag{7}$$

$$b_k = g_{x_{k+1}} - g_k \tag{8}$$

where g denotes the gradient at the time of iteration.

(II) The penalty factor λ is determined through K-fold cross-validation. Specifically, a sequence of candidate λ values is predefined, and for each candidate value the model is trained on $K-1$ folds and validated on the remaining fold. The mean squared error (MSE) is averaged over all K validation rounds, and the λ corresponding to the minimum average MSE is selected. The final Batch-Lasso model is then refitted using the selected λ.

3.2. Self-Adjustment of the Batch-Lasso Model

Relying solely on traditional Lasso may not ensure the model's effectiveness in the future, particularly in the event of significant changes in the distribution of meteorological factor data. Consequently, we propose a self-updating mechanism for model parameters, along with a model evaluation and rollback system based on the Batch approach.

Based on the results of weather multimode clustering, Batch-Lasso models are established for various modes. In the current implementation, the self-update mechanism is governed by a performance-verified update criterion rather than by an explicit two-sample distribution statistic such as the Kolmogorov–Smirnov test or Kullback–Leibler divergence. Specifically, when a candidate batch update is triggered, the new batch is incorporated into the corresponding mode dataset and a pre-updated Batch-Lasso model is re-estimated. The candidate updated model is accepted only if it achieves lower forecasting error on the verification batch than the previously accepted model; otherwise, the model parameters and forecasting results are rolled back to the last accepted state. Therefore, the rollback mechanism serves as a safeguard against ineffective updates rather than as a source of forecasting instability. The pseudocode for the Batch-Lasso model used in short-term PV power forecasting is presented in Algorithm 2.

In Algorithm 2, T denotes any specified time period, R represents the corresponding time scale, and D refers to the length and name of the dataset. The updated name of dataset D is denoted as D_{new}. Figure 2 illustrates the self-updating mechanism and evaluation rollback process of the Batch-Lasso method.

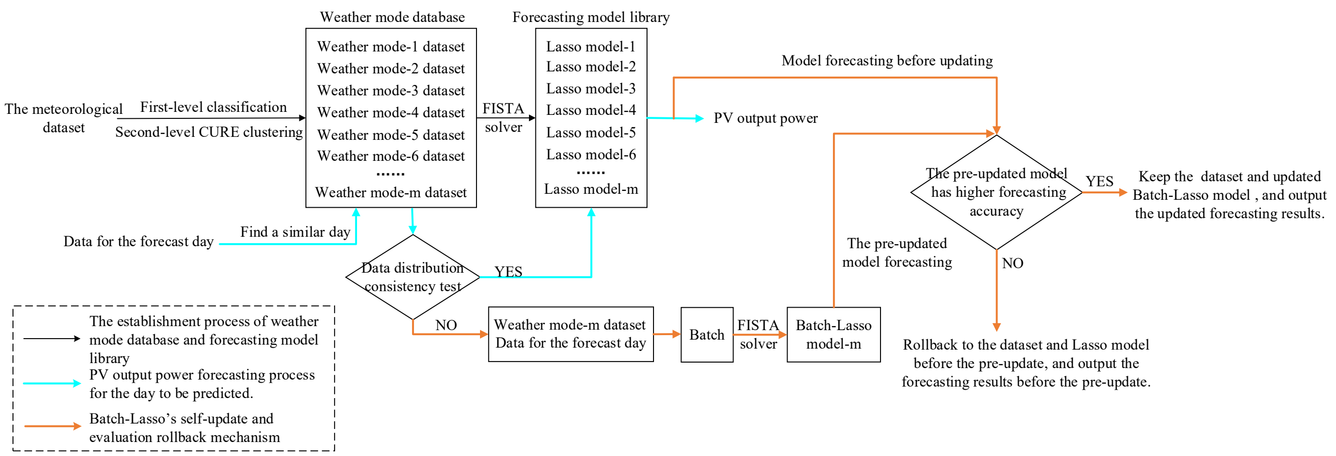


Figure 2. Flow chart of Batch-Lasso's self-updating and evaluation rollback mechanism and related processes.

Algorithm 2. The pseudocode of Batch-Lasso model for day-ahead short-term PV power forecasting process

Input: D Sets of Sample Data Before Period T
Output: PV Power Values from T to $T + R$
For $t = 0$ to R do
    Fit Lasso model 1 using the data for period $T + t$
    Predict the PV power values for period $T + t$
    Add the weather data and PV power forecasting values of the $T + t$ period to the D
    Remove the first set of data from D
    Set $D_{\text{new}} = D$
    Fit Lasso model 2 using D_{new} for period $T + t$
    Use model 1 and model 2 to predict PV power for period $T + t + 1$
    Compare the forecasting results of the two models
    Select the model with higher forecasting accuracy
    Use the selected model to predict the PV power for period $T + t + 2$
End for
Return PV power values from T to $T + R$

In Figure 2, the forecasting process for the designated forecast day is outlined as follows: (I) Identify days in the weather mode datasets that exhibit similarities to the forecast day; (II) Match the dataset associated with the identified similar day with the Lasso model; (III) Evaluate whether the distribution of meteorological data for the forecast day aligns with the data distribution of the weather mode dataset corresponding to the similar day. If the distributions are consistent, input the forecast day data into the Lasso model for forecasting; if not, integrate the forecast day data into the dataset and pre-update the Lasso model to create a Batch-Lasso model; (IV) Initiate the evaluation rollback mechanism for the Batch-Lasso model. If the prediction accuracy of the pre-updated Batch-Lasso model exceeds that of the previous version, retain the pre-updated model and its forecasting results; otherwise, rollback to the model and forecasting values prior to the pre-update.

The Batch approach enables the Lasso model to self-update and correct its parameters. The fitting coefficients of the Lasso model are dynamically adjusted based on the current data, which continuously improves the model's generalization ability for future forecasting while maintaining its predictive accuracy.

4. Implementation and Evaluation of Multimode Forecasting Models

The specific forecasting process is illustrated in Figure 3. First, the meteorological dataset is categorized into first-level modes based on seasonal variations. The resulting seasonal mode datasets are then further segmented into second-level modes using CURE, leading to the establishment of a multimode weather mode database. Second, a Batch-Lasso model is developed for each mode to facilitate short-term PV power forecasting. Third, the model's forecasting performance is evaluated following self-update procedures. If there is no improvement in forecasting accuracy after the pre-update, the dataset and forecasting model are rolled back to their state prior to the pre-update. Finally, historical datasets are utilized to conduct case studies, comparing the proposed model with other models.

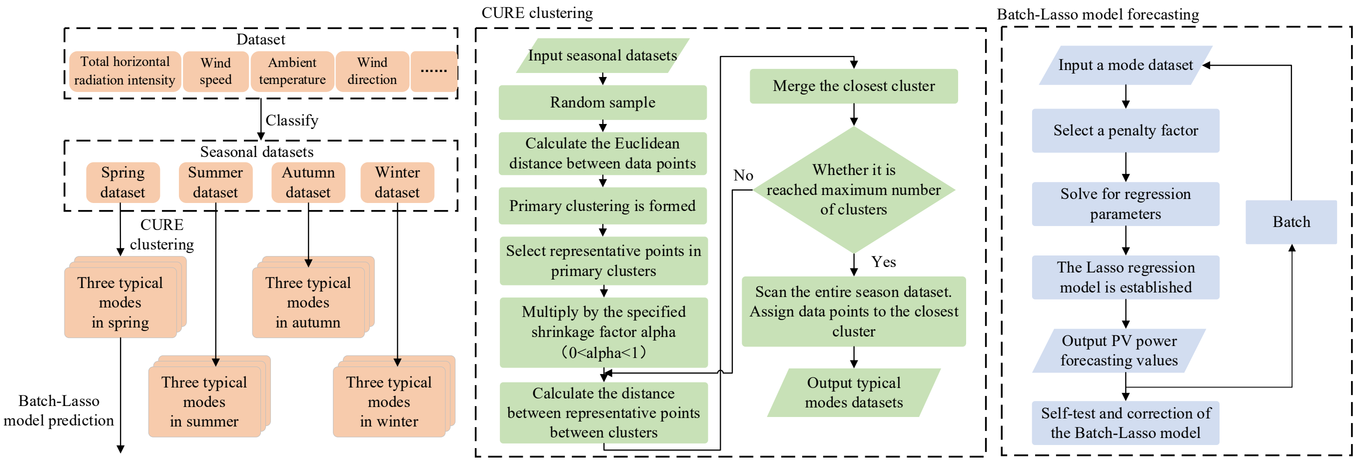


Figure 3. The multimode day-ahead short-term PV power forecasting framework, integrating CURE and Batch-Lasso.

5. Case Study

5.1. Dataset Quality and Preprocessing

This study focuses on the No. 2 power station within the Yulara solar system, located at the Desert Knowledge Australia Solar Center, which has a rated capacity of 226.8 kW [43]. The dataset comprises nine dimensions: wind speed (x_1), ambient temperature (x_2), total horizontal radiation intensity (x_3), wind direction (x_4), maximum wind speed (x_5), atmospheric pressure (x_6), pyranometer irradiance (x_7), external component temperature (x_8), and internal component temperature (x_9). Data were collected from April 2016 to December 2019. After removing invalid entries, a total of 179,829 data points remained, with a time resolution of 5 min. This dataset is referred to as Dataset A. We further applied CURE-based dual-dimension clustering to classify the data into weather modes. The first dimension categorizes the data by season (spring, summer, autumn, winter), while the second dimension identifies typical weather patterns within each season using the CURE algorithm. The number of weather types per season is adaptively determined by maximizing the Silhouette Score (k ∈ {2, 3, 4}). For the dataset utilized in this study, the optimal choice consistently corresponds to k = 2 across all seasons, resulting in a total of eight weather modes. For each modeling session, the data were divided by day, with a training-to-testing ratio of 3:1. In terms of the implementation environment, all experiments were conducted on a Windows 11 operating system, utilizing an Intel Core i5-12400 CPU (Intel Corporation, Santa Clara, CA, USA), 16 GB of RAM, and an NVIDIA GeForce RTX 5060 GPU (NVIDIA Corporation, Santa Clara, CA, USA).

Before establishing the Batch-Lasso model, the high-dimensional dataset of photovoltaic (PV) power undergoes a preprocessing phase, which is divided into two distinct steps. The first step involves the removal of outliers, the interpolation of missing values, and the alignment of data timestamps. In the second step, the dataset is normalized using the min-max normalization method. Notably, after the preprocessed dimensionless dataset is sent to CURE clustering, CURE is capable of further eliminating outliers.

Using the min-max normalization method, the calculation is described as follows:

$$q_i = \frac{p_i - \min\limits_{1 \le j \le n} \{p_j\}}{\max\limits_{1 \le j \le n} \{p_j\} - \min\limits_{1 \le j \le n} \{p_j\}} \tag{9}$$

where p is the original data series, q is the transformed dimensionless data series, n is the data column length.

5.2. Analysis of Multimode Clustering Results

The first-level modes are categorized according to traditional Chinese seasonal dates, while the second-level modes are classified using the CURE framework. The configuration of CURE is presented in Table 1, and the outcomes of the eight weather modes are illustrated in Figure 4.

Table 1. The parameters of CURE.

Number of Samples	The Number of Representative Points	Alpha	$l_{\max}$
3000	40	0.4	4

As illustrated in Figure 4, the primary weather patterns (WP1) across the four seasons exhibit similar bell-shaped trends with high amplitudes. In contrast, the secondary weather patterns (WP2) demonstrate significantly lower power output and greater variability between seasons. Therefore, it is essential to categorize the meteorological dataset into a multimode weather database based on both seasonal variations and the CURE clustering algorithm.

The sample sizes of the two weather modes in Dataset A are summarized in Appendix A Table A2. It is evident that WP2 contains substantially fewer samples than WP1 across all four seasons, with the WP2 subset comprising only approximately 1% to 6% of the corresponding WP1 subset. This observation confirms that WP2 represents a distinctly data-limited mode. In principle, weighted regression or resampling could be considered for such smaller subsets. However, these strategies were not adopted in the present study, as the proposed framework constructs separate models for each weather mode rather than fitting a single model to a pooled imbalanced dataset. Instead, the smaller WP2 modes are addressed through mode-specific modeling and sparse Batch-Lasso estimation.

To quantitatively assess the suitability of CURE for the present dataset, a clustering ablation study was conducted on Dataset A, maintaining consistent seasonal partitions. CURE was compared with K-means and GMM using the Silhouette Score and Davies–Bouldin Index; the corresponding runtimes were also recorded to evaluate processing costs. The results are summarized in Table 2.

Table 2. Clustering validity and runtime comparison.

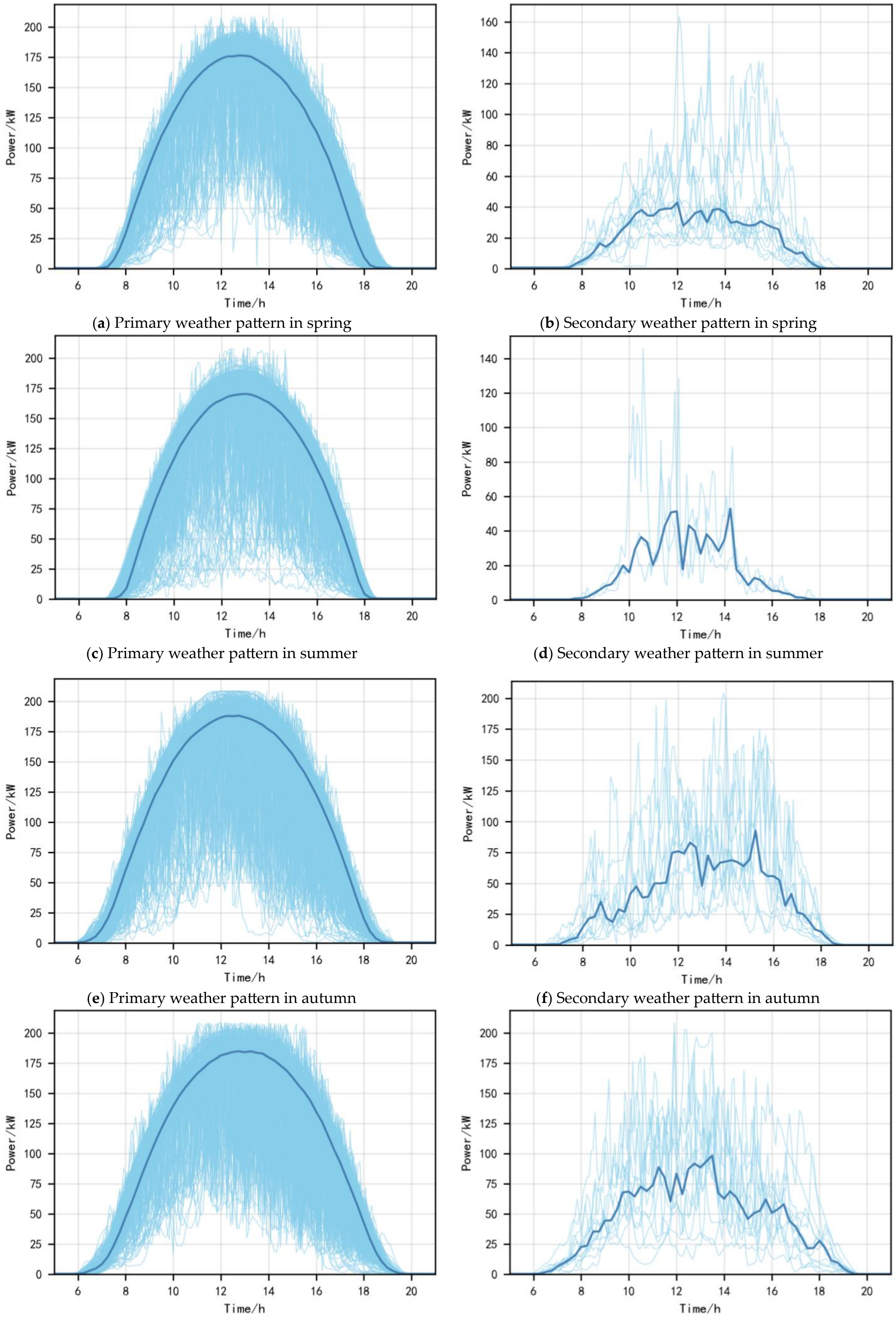
Method	Avg Silhouette	Avg DB	Avg Runtime per Season (ms)
K-means	0.315	1.381	16.9
GMM	0.180	2.054	47.3
CURE	0.418	0.814	37.0

As shown in Table 2, CURE achieved the highest average Silhouette Score and the lowest average Davies–Bouldin Index on Dataset A. This indicates that CURE produced more compact and better-separated weather-mode structures compared to K-means and GMM. Furthermore, although CURE was not the fastest clustering method, its runtime remained within the millisecond scale. These results substantiate the use of CURE in this study primarily as a weather-mode structuring tool with enhanced clustering validity.

5.3. Model Establishment

In the establishment of the Batch-Lasso model, a 10-fold cross-validation approach is utilized to determine the optimal value of λ. For each Lasso regression, 200 distinct values of λ are assessed, with the λ that results in the lowest cross-validation mean squared error (MSE) being selected. The Batch-Lasso models corresponding to these eight configurations are presented in Table 3.

https://doi.org/10.3390/su18073319



(**a**) Primary weather pattern in spring (**b**) Secondary weather pattern in spring

(**c**) Primary weather pattern in summer (**d**) Secondary weather pattern in summer

(**e**) Primary weather pattern in autumn (**f**) Secondary weather pattern in autumn

(**g**) Primary weather pattern in winter (**h**) Secondary weather pattern in winter

Figure 4. Clustering results of 8 weather modes.

Table 3. Batch-Lasso models in 8 modes.

Batch-Lasso	WP1	WP2
Spring	$y_1 = -0.101x_2 + 0.178x_3 + 0.022x_5 + 0.047x_6 + 1.067x_7$	$y_2 = 0.694x_3 + 0.756x_7$
Summer	$y_3 = 0.016x_1 - 0.057x_2 + 0.599x_3 - 0.012x_4 + 0.015x_5 + 0.841x_7 - 0.010x_8$	$y_4 = -0.094x_2 + 0.714x_3 - 0.014x_6 + 0.848x_7 + 0.081x_8 + 0.037x_9$
Autumn	$y_5 = 0.025x_1 - 0.178x_2 + 0.531x_3 + 0.016x_5 + 0.040x_6 + 0.688x_7 + 0.089x_9$	$y_6 = 0.691x_3 + 0.014x_5 + 0.687x_7$
Winter	$y_7 = 0.015x_1 - 0.105x_2 + 0.576x_3 + 0.013x_5 + 0.589x_7 + 0.043x_9$	$y_8 = 0.632x_3 + 0.655x_7$

In Table 3, y refers to the PV power of forecasting, while x indicates the feature variables. An analysis of Table 3 reveals that for primary weather patterns characterized by larger training samples, the Batch-Lasso model retains a greater number of meteorological variables, specifically ranging from 5 to 7 features. In contrast, for secondary weather patterns with smaller sample sizes, the model selects fewer variables, typically between 2 and 3 features, with a predominant reliance on radiation-related features (x_3, x_7). This demonstrates the Lasso regularization's capability to automatically achieve sparser representations under data-limited conditions.

From a physical perspective, the selected features align with the operational mechanisms of PV systems. Radiation-related variables, specifically total horizontal radiation intensity (x_3) and pyranometer irradiance (x_7), are retained across all eight weather modes and typically exhibit the largest coefficients. This indicates that solar irradiance is the predominant driver of PV power generation throughout the seasons. This observation is physically justified, as the output current of a PV system is primarily influenced by the amount of incident solar energy. In contrast, temperature- and wind-related variables play a secondary, yet season-dependent, role. Ambient temperature (x_2) is present in most modes and generally shows a negative coefficient, which is consistent with the established reduction in PV conversion efficiency at elevated cell temperatures. Wind speed (x_1), maximum wind speed (x_5), and atmospheric pressure (x_6) are retained only in certain seasons or primary weather patterns, suggesting that their influence is primarily indirect, for instance, through convective cooling and weather stability. Overall, the seasonal variation in selected features reflects the combined effects of irradiance availability, thermal conditions, and weather-dependent operating regimes.

To verify the validity of the constructed models, we performed ordinary least squares (OLS) regression using the selected features for each model group and conducted a standard t-test on each regression coefficient. The null hypothesis is denoted as H_0: $\beta_i = 0$, i.e., which states that the selected meteorological variables have no linear contribution to PV power output. The results are presented in Table 4.

Table 4. The p-values from the t-tests for the 8 model groups.

p	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9
y_1	/	0.0000	0.0000	/	0.0000	0.0000	0.0000	/	/
y_2	/	/	0.000	/	/	/	0.0000	/	/
y_3	0.0000	0.0000	0.0000	0.0000	0.7377	/	0.0000	0.0000	/
y_4	/	0.5945	0.0000	/	/	0.0027	0.0000	0.2454	0.7113
y_5	0.0000	0.0000	0.0000	/	0.0013	0.0000	0.0000	/	0.8894
y_6	/	/	0.0000	/	0.0357	/	0.0003	/	/
y_7	0.0000	0.0000	0.0000	/	0.0001	/	0.0000	/	0.0000
y_8	/	/	0.7930	/	/	/	0.0000	/	/

A careful examination of the p-values in Table 4 reveals that the majority of p-values are less than 0.05. Consequently, we reject the null hypothesis for these variables, confirming that the meteorological factors selected in the model groups are significantly linearly related to PV power output, thereby validating the constructed models for linear regression prediction.

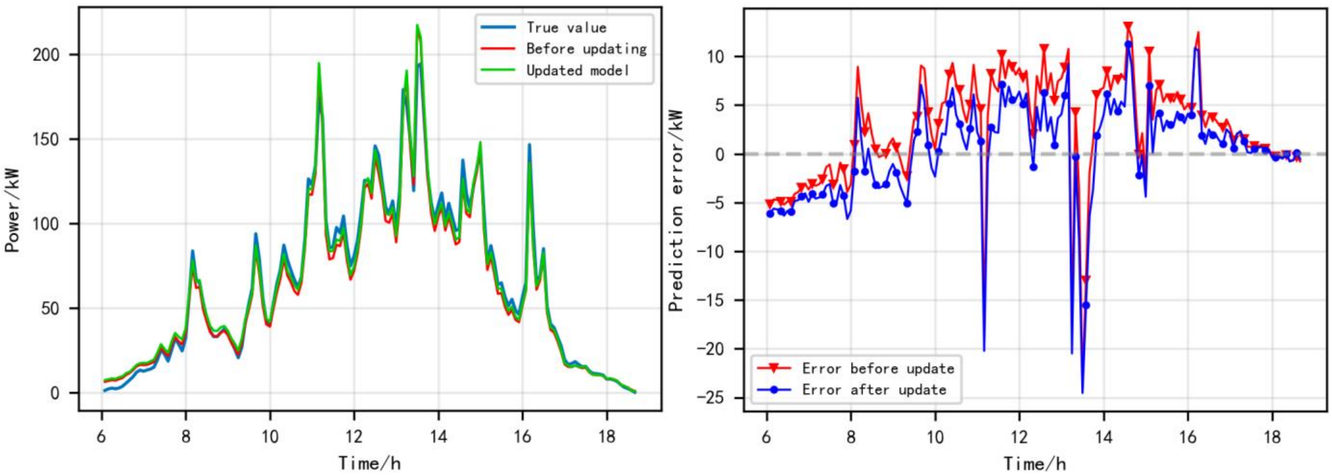
5.4. Analysis of Forecasting Results

5.4.1. Validation of the Effectiveness of Self-Updating Mechanisms

We employ an extreme scenario to assess the true impact of the Batch-based self-update capability on the accuracy of PV power forecasting. The model expressions before and after the pre-update are presented in Table 5. The errors associated with short-term PV power forecasting results, both prior to and following the pre-update, are displayed in Table 8. The forecasting results of the Batch-Lasso model, both before and after the pre-update, under typical weather modes are illustrated in Figure 5a, while the corresponding forecasting errors are shown in Figure 5b.

Table 5. Models before and after the pre-update.

Before pre-update	$y_{\text{ep}} = -0.101x_2 + 0.178x_3 + 0.022x_5 + 0.047x_6 + 1.067x_7$
After pre-update	$y_{\text{ep-pre}} = -0.156x_2 + 0.514x_3 + 0.037x_5 + 0.092x_6 + 0.713x_7 + 0.011x_8 + 0.078x_9$



(**a**) Fitted Curve Comparison (**b**) Error Curve Comparison

Figure 5. Comparison of forecasting results and errors before and after model pre-update.

Table 5 presents the forecasting models before and after the pre-update. It is evident that, compared to the original model, the pre-updated model has significantly strengthened the coefficient of variable x_3 while reducing the dominance of x_7. Additionally, the pre-updated model introduces variables x_8 and x_9, which were previously compressed to zero by Lasso regularization. This improvement is attributed to the pre-updated model utilizing a dataset that is more relevant to the forecast day, allowing it to recover informative features that were suppressed during the initial training phase. Therefore, it can be concluded that combining the Batch approach with the Lasso model results in a model that adaptively adjusts to evolving weather conditions while maintaining prediction accuracy.

An analysis of the data presented in Table 6 and Figure 5 reveals that the self-updating Batch-Lasso model demonstrates reduced forecasting errors and improved forecasting accuracy compared to the model prior to the pre-update. This finding substantiates the effectiveness of the self-updating mechanism based on the Batch approach.

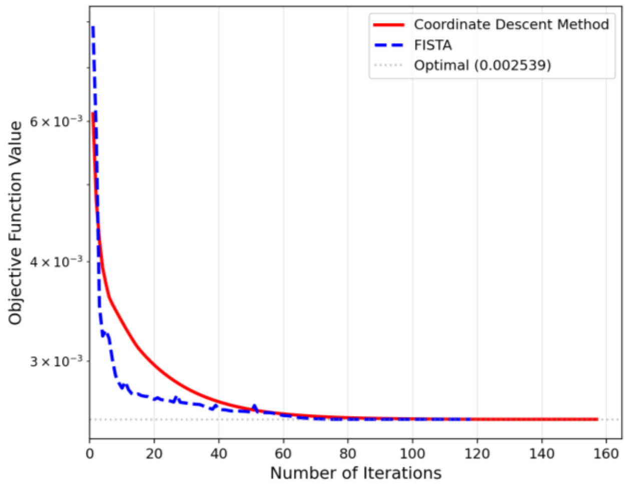
Table 6. Errors of short-term PV power forecasting results before and after model pre-update.

	R^2	MAE/kW	MSE/kW^2	MAPE
Before pre-update	0.9838	4.845	38.109	7.835%
After pre-update	0.9881	3.877	28.020	7.299%

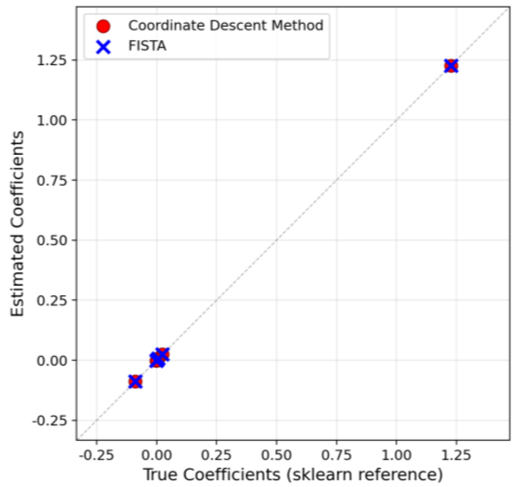
Rollback is more likely to occur when newly arrived batches reflect abrupt weather changes or atypical operating conditions, which can impede the immediate improvement of verification performance following a model refresh. In these instances, the model reverts to its last accepted state. Consequently, rollback functions as a protective mechanism against ineffective updates, rather than as a direct contributor to forecasting instability.

5.4.2. Validation of the Superiority of the FISTA Algorithm

In the parameter estimation process of the Lasso regression model, the efficiency and stability of the optimization algorithm directly influence model construction and the utilization of computational resources. This subsection compares the FISTA algorithm introduced in Section 3.1 with the coordinate descent method, both employing the same regularization parameter λ and convergence tolerance. As illustrated in Figure 6a, FISTA exhibits significantly faster convergence, reaching the optimal objective value in fewer iterations compared to coordinate descent. Figure 6b further confirms that both methods converge to identical regression coefficients, indicating that FISTA achieves its speed advantage without compromising solution quality. These findings validate the effectiveness and superiority of FISTA in estimating Lasso regression parameters within the proposed framework.



(**a**) Fitted Curve Comparison



(**b**) Error Curve Comparison

Figure 6. Performance comparison of regression parameter estimation algorithm.

The analysis of Figure 6a, which compares the convergence rates of FISTA and coordinate descent, reveals that the FISTA algorithm exhibits significantly accelerated convergence characteristics. In the initial iteration phase, FISTA achieves a substantial reduction in the objective function error compared to coordinate descent, with its convergence curve displaying a smooth exponential decay trend. While coordinate descent begins converging only at the fourth iteration, FISTA reaches the preset convergence threshold by the second iteration, effectively reducing computational time consumption.

As observed in Figure 6b, which compares the parameter estimation results between FISTA and gradient descent, both algorithms ultimately yield regression parameter vectors that are identical within numerical precision. This finding validates that FISTA maintains solution uniqueness and accuracy while optimizing the iteration path through Nesterov's historical gradient information. The optimization arises from the corrective influence of the inertial term on the gradient direction in Nesterov acceleration.

In summary, when solving for Lasso regression parameters, FISTA demonstrates superior performance compared to traditional gradient descent methods. Specifically, FISTA exhibits smaller initial errors, faster descent rates, and requires fewer iterations. Collectively, these results validate the effectiveness and superiority of the FISTA algorithm.

5.4.3. Comparison with Conventional Regression Baselines

To validate the superiority of the proposed Batch-Lasso framework in resource-constrained edge scenarios, we selected four classical, computation-efficient regression algorithms as baselines: Standard Lasso, classical linear regression, ridge regression, and least angle regression (LARS). Unlike deep learning models, these algorithms exemplify the state-of-the-art for lightweight deployment.

We randomly select one representative day from eight weather modes for day-ahead short-term PV power forecasting. Figure 7 visualizes the fitting trajectories of these five models against the actual power generation curves.

Tables 7–11 present the forecasting errors of the Batch-Lasso model in comparison with four other regression models across the eight weather modes. Additionally, we report the global error metrics calculated over the entire test set to enable a comprehensive performance comparison.

Table 7. Forecasting errors of regression baselines for Spring weather modes.

	Indicators	Batch-Lasso	Lasso	Ridge	Classical Linear	Least Angle
WP1	R^2	0.9966	0.9879	0.9870	0.9870	0.9874
	MAE/kW	2.755	4.579	4.924	4.924	4.782
	MSE/kW^2	11.657	41.430	44.694	44.694	43.226
	MAPE/%	4.64	10.01	10.87	10.87	10.64
WP2	R^2	0.9940	0.9938	0.9952	0.9952	0.9938
	MAE/kW	1.636	1.671	1.492	1.491	1.665
	MSE/kW^2	6.800	7.017	5.389	5.380	6.948
	MAPE/%	0.0882	4.00	3.83	3.83	4.23

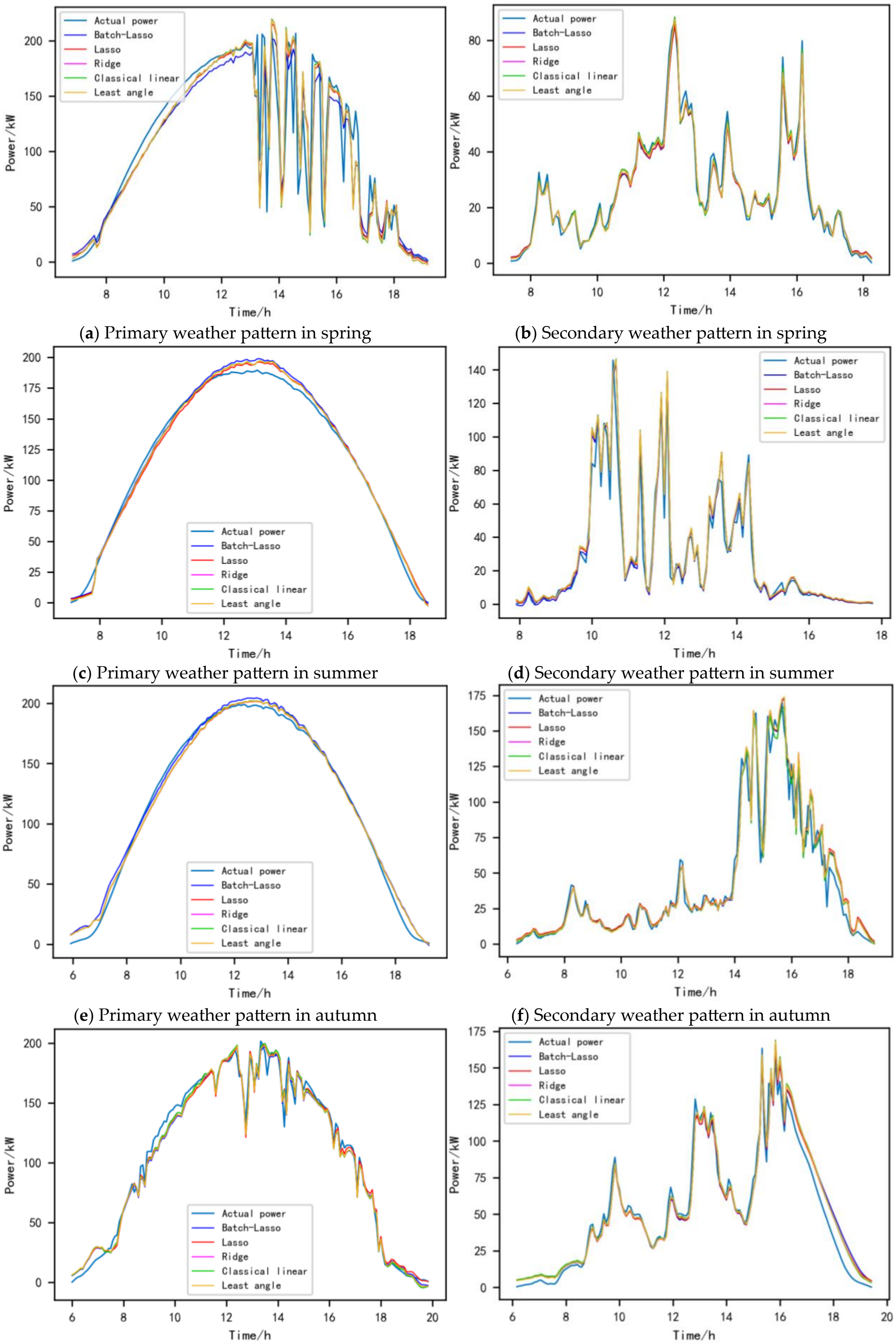
Table 8. Forecasting errors of regression baselines for summer weather modes.

	Indicators	Batch-Lasso	Lasso	Ridge	Classical Linear	Least Angle
WP1	R^2	0.9945	0.9922	0.9943	0.9943	0.9942
	MAE/kW	3.524	4.375	3.533	3.533	3.558
	MSE/kW^2	17.072	24.041	17.713	17.713	17.893
	MAPE/%	5.86	7.13	6.09	6.09	6.14
WP2	R^2	0.9545	0.9503	0.9493	0.9493	0.9499
	MAE/kW	3.916	4.040	4.268	4.268	4.219
	MSE/kW^2	50.422	55.028	56.114	56.120	55.507
	MAPE/%	16.15	16.95	17.44	17.44	17.37

Table 9. Forecasting errors of regression baselines for Autumn weather modes.

	Indicators	Batch-Lasso	Lasso	Ridge	Classical Linear	Least Angle
WP1	R^2	0.9949	0.9931	0.9939	0.9939	0.9935
	MAE/kW	3.439	4.159	3.846	3.846	4.006
	MSE/kW^2	17.891	24.481	21.486	21.486	23.061
	MAPE/%	4.90	5.97	5.57	5.57	5.73
WP2	R^2	0.9852	0.9678	0.9762	0.9762	0.9709
	MAE/kW	4.609	7.623	5.957	5.955	7.199
	MSE/kW^2	32.604	70.837	52.230	52.273	63.952
	MAPE/%	14.79	21.52	18.48	18.48	20.41

https://doi.org/10.3390/su18073319



(**a**) Primary weather pattern in spring (**b**) Secondary weather pattern in spring
(**c**) Primary weather pattern in summer (**d**) Secondary weather pattern in summer
(**e**) Primary weather pattern in autumn (**f**) Secondary weather pattern in autumn
(**g**) Primary weather pattern in winter (**h**) Secondary weather pattern in winter

Figure 7. Forecasting comparison with regression baselines under 8 weather modes.

Table 10. Forecasting errors of regression baselines for Winter weather modes.

	Indicators	Batch-Lasso	Lasso	Ridge	Classical Linear	Least Angle
WP1	R^2	0.9937	0.9936	0.9924	0.9924	0.9934
	MAE/kW	3.874	3.848	4.197	4.197	3.916
	MSE/kW^2	21.977	22.401	26.804	26.805	23.157
	MAPE/%	6.14	6.81	5.76	5.76	5.75
WP2	R^2	0.9847	0.9700	0.9726	0.9726	0.9758
	MAE/kW	1.594	2.783	2.808	2.811	2.520
	MSE/kW^2	6.916	13.500	12.344	12.360	10.896
	MAPE/%	6.05	8.07	8.85	8.86	7.51

Table 11. Forecasting errors of regression baselines on the entire test set.

Indicators	Batch-Lasso	Lasso	Ridge	Classical Linear	Least Angle
R^2	0.9669	0.9632	0.9639	0.9639	0.9639
MAE/kW	6.854	7.501	7.128	7.128	7.182
MSE/kW^2	133.645	148.433	145.404	145.404	145.439
MAPE/%	9.82	11.04	10.67	10.67	10.72

An examination of Tables 7–10 indicates that Batch-Lasso matches or outperforms standard regression baselines in weather-dependent modes, with the most significant gains observed under the more variable secondary patterns. While all linear models exhibit strong performance for stable primary patterns, Batch-Lasso sustains its accuracy as conditions become less regular. This robustness extends to the entire test set. As illustrated in Table 12, Batch-Lasso achieves the best overall regression performance, surpassing Lasso, ridge regression, classical linear regression, and least angle regression.

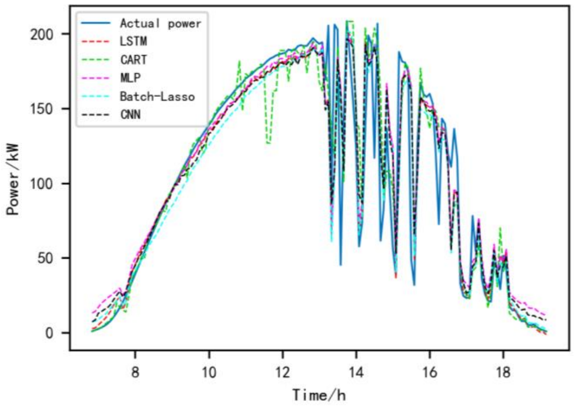
Table 12. Forecasting errors of machine learning baselines for Summer weather modes.

	Indicators	Batch-Lasso	LSTM	CNN	MLP	CART
WP1	R^2	0.9945	0.9927	0.9911	0.9876	0.9793
	MAE/kW	3.524	4.275	4.426	5.757	5.478
	MSE/kW^2	17.072	22.491	27.563	38.437	64.050
	MAPE/%	5.86	5.89	7.30	6.79	5.59
WP2	R^2	0.9545	0.9411	0.7457	0.9043	0.8016
	MAE/kW	3.916	5.550	14.274	7.102	6.787
	MSE/kW^2	50.422	65.256	281.635	105.990	219.779
	MAPE/%	16.15	21.58	48.01	29.81	21.05

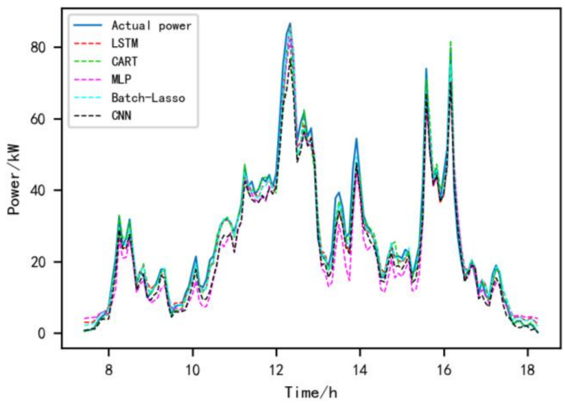
5.4.4. Comparison with Machine Learning Baselines and Computational Cost

Building on the regression-only comparison presented in Section 5.4.3, we further benchmark Batch-Lasso against several machine learning baselines to evaluate the resource requirements of various models while maintaining comparable forecasting accuracy. This analysis highlights the lightweight and efficient characteristics of Batch-Lasso for edge deployment. The baselines include deep learning models (LSTM, CNN, and MLP) as well as a tree-based model (CART). The representative-day forecasting comparisons under the eight weather modes are illustrated in Figure 8, and the key hyperparameter settings for the deep learning baselines are provided in Appendix A Table A3.

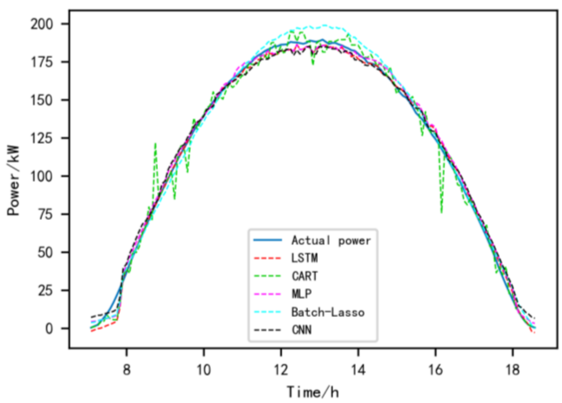
To quantify these visual differences, the detailed error statistics for WP1 and WP2 in each season are reported in Tables 12–16.



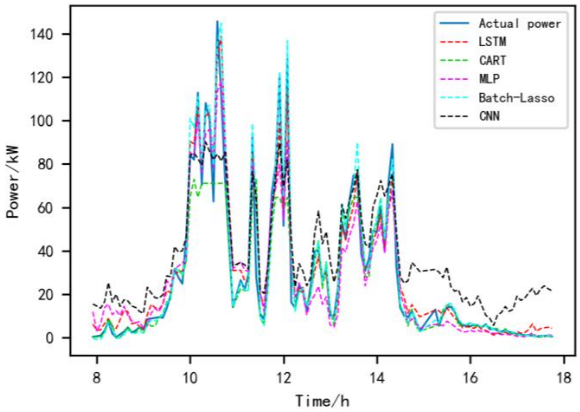
(**a**) Primary weather pattern in spring



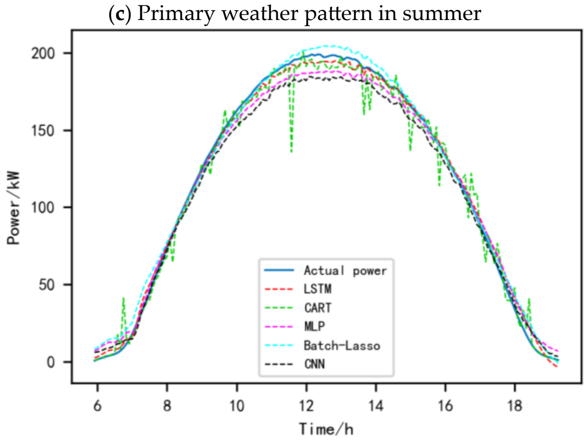
(**b**) Secondary weather pattern in spring



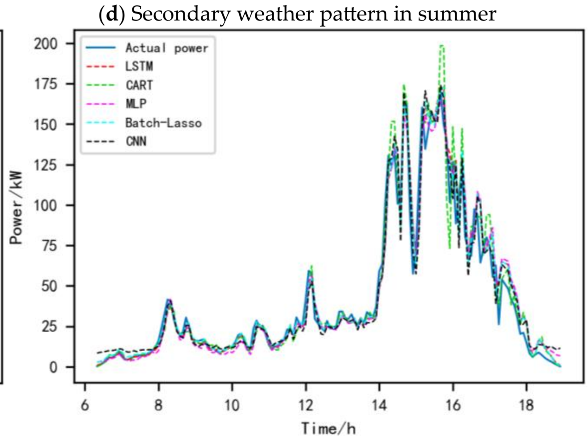
(**c**) Primary weather pattern in summer



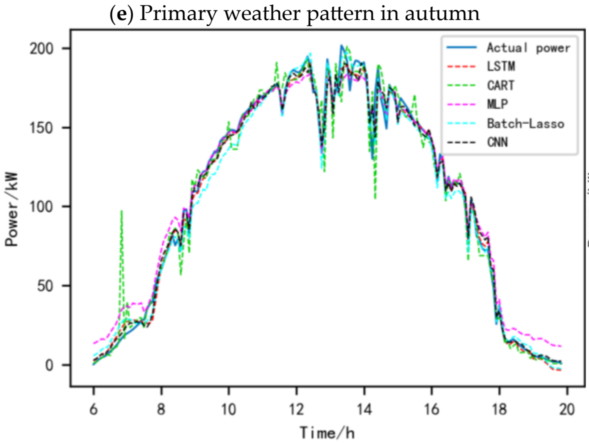
(**d**) Secondary weather pattern in summer



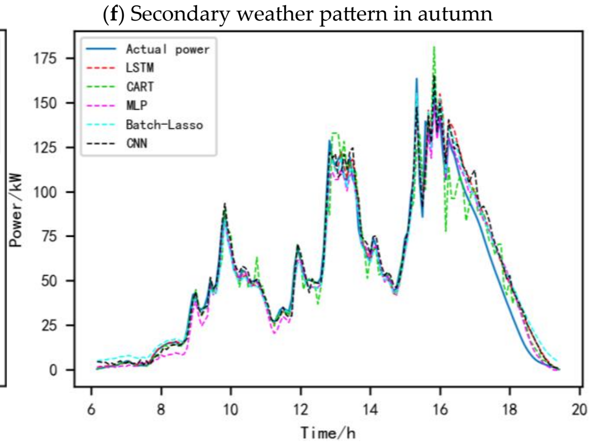
(**e**) Primary weather pattern in autumn



(**f**) Secondary weather pattern in autumn



(**g**) Primary weather pattern in winter



(**h**) Secondary weather pattern in winter

Figure 8. Forecasting comparison with machine learning baselines under 8 weather modes.

Table 13. Forecasting errors of machine learning baselines for Spring weather modes.

	Indicators	Batch-Lasso	LSTM	CNN	MLP	CART
WP1	R^2	0.9966	0.9949	0.9941	0.9906	0.9937
	MAE/kW	2.755	3.487	3.515	4.897	3.155
	MSE/kW^2	11.657	17.455	20.312	32.338	21.544
	MAPE/%	4.64	5.28	5.94	8.14	4.67
WP2	R^2	0.9940	0.9881	0.9730	0.9817	0.9590
	MAE/kW	1.636	2.139	3.391	3.453	3.821
	MSE/kW^2	6.800	13.362	30.401	20.595	46.071
	MAPE/%	3.86	4.38	8.71	9.75	8.77

Table 14. Forecasting errors of machine learning baselines for Autumn weather modes.

	Indicators	Batch-Lasso	LSTM	CNN	MLP	CART
WP1	R^2	0.9949	0.9968	0.9897	0.9944	0.9857
	MAE/kW	3.439	2.434	5.188	3.644	4.789
	MSE/kW^2	17.891	11.452	36.406	19.766	50.467
	MAPE/%	4.90	4.43	6.05	5.07	7.34
WP2	R^2	0.9852	0.9891	0.9653	0.9887	0.7383
	MAE/kW	4.609	4.214	6.738	4.020	14.598
	MSE/kW^2	32.604	23.896	76.198	24.846	575.236
	MAPE/%	14.79	11.57	12.75	12.30	26.77

Table 15. Forecasting errors of machine learning baselines for Winter weather modes.

	Indicators	Batch-Lasso	LSTM	CNN	MLP	CART
WP1	R^2	0.9937	0.9970	0.9951	0.9891	0.9786
	MAE/kW	3.874	2.702	3.301	5.114	4.891
	MSE/kW^2	21.977	10.674	17.062	38.194	75.029
	MAPE/%	6.14	3.62	4.73	9.58	6.67
WP2	R^2	0.9847	0.9778	0.9410	0.9277	0.9647
	MAE/kW	1.594	2.421	3.192	4.853	2.234
	MSE/kW^2	6.916	10.006	26.609	32.592	15.889
	MAPE/%	6.05	7.40	10.40	24.70	7.75

Table 16. Forecasting errors of machine learning baselines on the entire test set.

Indicators	Batch-Lasso	LSTM	CNN	MLP	CART
R^2	0.9669	0.9737	0.9738	0.9731	0.9524
MAE/kW	6.854	5.242	5.274	5.863	6.790
MSE/kW^2	133.645	105.958	105.559	108.517	191.980
MAPE/%	9.82	7.75	7.80	8.51	9.64

Tables 12–15 illustrate a pronounced dependence of forecasting performance on the mode, with more significant discrepancies observed in the secondary patterns. The substantially fewer days in WP2, compared to WP1, result in models such as LSTM, CNN, and MLP exhibiting a higher propensity for overfitting and performance fluctuations when trained with limited within-mode data. Conversely, Batch-Lasso achieves the lowest MAE/MSE in five of the eight modes, indicating superior data efficiency and robustness in small-sample contexts. The L1 regularization employed promotes sparse representations and effectively controls complexity on a per-mode basis. Moreover, the roll-back-enabled batch self-update mechanism accepts updates only when they demonstrate verified improvements, thereby mitigating accuracy deterioration in data-scarce modes. The global results presented in Table 16 further suggest that while deep learning models attain better average metrics across the entire test set, their stability is compromised in small-sample modes similar to WP2. This observation necessitates the computational overhead comparison illustrated in Table 17, which evaluates the accuracy–efficiency trade-off for edge deployment of ICTs.

Table 17. Computational overhead of machine learning baselines.

Indicators	Batch-Lasso	LSTM	CNN	MLP	CART
Params	53	169,001	80,513	9601	268,435
Train(s)	0.902	85.129	150.223	128.054	1.239
Infer(s)	0.877	0.035	0.100	0.022	0.012
Mem(MB)	0.10	39.42	30.88	29.79	6.18

Table 17 highlights significant differences in computational overhead across various models. Batch-Lasso is the most lightweight model, featuring only 53 parameters, a memory footprint of 0.10 MB, and a training time of 0.902 s, indicating minimal storage and computational requirements. In contrast, LSTM, CNN, and MLP exhibit substantially higher model sizes and memory footprints, with approximately 9.6k to 169k parameters and memory usage ranging from 29.79 to 39.42 MB. These models also require considerably longer training times, ranging from 85.129 to 150.223 s, reflecting significantly higher training and storage costs. Although CART achieves the fastest inference time at 0.012 s, its parameter count and memory usage are notably larger than those of Batch-Lasso, with 268,435 parameters and 6.18 MB of memory. Overall, while deep learning baselines offer improved accuracy, they come with markedly increased resource demands. In contrast, Batch-Lasso maintains competitive accuracy while substantially reducing model size and memory footprint, making it more suitable for resource-constrained edge deployment.

To further validate the observed performance differences, supplementary significance analyses were conducted on Dataset A for the overall benchmark comparison. Specifically, the Diebold–Mariano (DM) test and the Wilcoxon signed-rank test were employed, and 95% bootstrap confidence intervals for the Mean Absolute Error (MAE) were computed. The results indicate that Batch-Lasso achieves statistically significant improvements over the primary linear baselines, while its performance relative to CART and MLP remains statistically comparable. Additionally, LSTM and CNN exhibit lower overall MAE on Dataset A. Detailed results can be found in Appendix A (Tables A4 and A5).

In addition, a supplementary experiment was conducted in which LSTM, CNN, and MLP were trained using the pooled seasonal datasets and then tested separately under different weather modes. Specifically, unlike the main experiments where the deep learning baselines were trained and tested within each individual mode (i.e., WP1-trained/WP1-tested and WP2-trained/WP2-tested), the supplementary setting used the entire seasonal dataset for training and then evaluated the trained models separately on WP1 and WP2. Detailed results are provided in Appendix A (Table A6). This supplementary experiment provides an additional comparative perspective, but the main analysis in this study remains based on the mode-specific forecasting setting.

5.5. External Validation and Cross-Site Evaluation

To assess the transferability of the proposed framework beyond the Australian site, we introduce an additional PV dataset collected from a 20 MW photovoltaic power station located in Hebei Province [44], China (denoted as Dataset B). This dataset includes PV power measurements alongside meteorological variables, including global horizontal irradiance (GHI), diffuse horizontal irradiance (DHI), ambient temperature, surface pressure, wind direction, and wind speed. Data were collected from 30 June 2018, to 12 June 2019, with a sampling interval of 15 min. After removing invalid entries, a total of 15,374 data points remained for subsequent analysis. Dataset B contains the same type of meteorological inputs and PV power measurements, but differs in local climate characteristics and operational conditions, providing a practical external validation setting. We apply the same preprocessing, multimode partitioning, and evaluation protocol as described in the previous section. Mode-wise errors for the seasonal weather modes are reported in Tables 18–21, while global test-set performance is summarized in Table 22, and computational overhead is compared in Table 23.

Table 18. Forecasting errors of all models for Spring weather modes in Dataset B.

	Indicators	Batch-Lasso	Lasso	Ridge	Classical Linear	Least Angle	LSTM	CNN	MLP	CART
WP1	R^2	0.9860	0.9853	0.9848	0.9848	0.9849	0.9896	0.9877	0.9950	0.9680
	MAE/kW	0.462	0.477	0.485	0.485	0.484	0.403	0.359	0.239	0.550
	MSE/kW^2	0.323	0.338	0.349	0.349	0.347	0.239	0.283	0.114	0.736
	MAPE/%	7.92	11.40	11.60	11.60	11.58	9.35	5.60	4.18	8.55
WP2	R^2	0.9911	0.9848	0.9854	0.9851	0.9947	0.9901	0.9163	0.9562	0.9827
	MAE/kW	0.086	0.112	0.161	0.163	0.087	0.124	0.352	0.193	0.143
	MSE/kW^2	0.020	0.034	0.032	0.033	0.012	0.022	0.185	0.097	0.038
	MAPE/%	3.21	4.14	8.47	8.59	4.25	5.99	14.76	7.82	6.94

Table 19. Forecasting errors of all models for Summer weather modes in Dataset B.

	Indicators	Batch-Lasso	Lasso	Ridge	Classical Linear	Least Angle	LSTM	CNN	MLP	CART
WP1	R^2	0.9919	0.9911	0.9906	0.9906	0.9907	0.9926	0.9624	0.9873	0.9727
	MAE/kW	0.185	0.200	0.209	0.209	0.208	0.194	0.457	0.303	0.263
	MSE/kW^2	0.070	0.077	0.081	0.081	0.081	0.064	0.326	0.110	0.237
	MAPE/%	6.88	7.05	6.92	6.92	6.94	6.36	14.23	10.01	8.62
WP2	R^2	0.9820	0.9805	0.9840	0.9818	0.9757	0.9809	0.9702	0.8922	0.9544
	MAE/kW	0.111	0.113	0.098	0.111	0.126	0.107	0.140	0.274	0.163
	MSE/kW^2	0.018	0.020	0.016	0.018	0.025	0.019	0.030	0.109	0.046
	MAPE/%	4.95	5.60	3.16	4.28	6.40	3.29	5.57	5.69	10.94

Table 20. Forecasting errors of all models for Autumn weather modes in Dataset B.

	Indicators	Batch-Lasso	Lasso	Ridge	Classical Linear	Least Angle	LSTM	CNN	MLP	CART
WP1	R^2	0.9919	0.9880	0.9878	0.9878	0.9878	0.9903	0.9653	0.9702	0.9534
	MAE/kW	0.395	0.483	0.489	0.489	0.489	0.430	0.781	0.752	0.766
	MSE/kW^2	0.222	0.328	0.335	0.335	0.334	0.265	0.952	0.818	1.278
	MAPE/%	6.84	9.02	9.11	9.11	9.10	7.46	8.34	8.79	10.33
WP2	R^2	0.9700	0.9566	0.9428	0.9422	0.9510	0.9545	0.8853	0.8464	0.9493
	MAE/kW	0.218	0.203	0.278	0.281	0.243	0.222	0.479	0.557	0.250
	MSE/kW^2	0.081	0.117	0.154	0.156	0.132	0.123	0.309	0.414	0.137
	MAPE/%	8.37	6.27	9.75	9.87	8.29	6.91	16.35	21.98	9.03

Table 21. Forecasting errors of all models for Winter weather modes in Dataset B.

	Indicators	Batch-Lasso	Lasso	Ridge	Classical Linear	Least Angle	LSTM	CNN	MLP	CART
WP1	R^2	0.9472	0.9463	0.9845	0.9848	0.9807	0.6031	0.8609	0.6124	0.9229
	MAE/kW	1.071	1.078	0.568	0.564	0.634	2.920	1.796	2.967	1.229
	MSE/kW^2	1.540	1.567	0.452	0.445	0.561	11.573	4.055	11.301	2.247
	MAPE/%	11.22	11.19	6.98	6.95	7.47	29.67	20.00	31.60	12.77
WP2	R^2	0.9943	0.9942	0.9940	0.9940	0.9940	0.9943	0.9895	0.9653	0.9772
	MAE/kW	0.211	0.214	0.209	0.209	0.211	0.229	0.307	0.470	0.410
	MSE/kW^2	0.075	0.076	0.078	0.078	0.079	0.075	0.137	0.453	0.297
	MAPE/%	5.40	5.44	5.66	5.66	5.62	6.09	6.17	14.38	7.97

Table 22. Forecasting errors of all models on the entire test set in Dataset B.

Indicators	Batch-Lasso	Lasso	Ridge	Classical Linear	Least Angle	LSTM	CNN	MLP	CART
R^2	0.9502	0.8975	0.8973	0.8973	0.8973	0.9397	0.9561	0.9533	0.9173
MAE/kW	0.835	1.358	1.361	1.361	1.360	0.873	0.741	0.760	0.966
MSE/kW^2	1.590	3.272	3.278	3.278	3.277	1.925	1.400	1.492	2.640
MAPE/%	16.09	26.81	26.86	26.86	26.86	17.54	13.32	13.88	16.87

https://doi.org/10.3390/su18073319

Table 23. Computational overhead of machine learning baselines in Dataset B.

Indicators	Batch-Lasso	LSTM	CNN	MLP	CART
Params	44	166,601	55,937	9217	20,913
Train(s)	0.054	15.671	18.786	11.816	0.043
Infer(s)	0.051	0.005	0.012	0.003	0.001
Mem(MB)	0.10	35.07	20.37	18.26	0.41

Tables 18–21 illustrate that the mode-wise results on Dataset B exhibit significant performance heterogeneity across different weather modes. During the spring, summer, and autumn (WP1/WP2) seasons, Batch-Lasso is generally comparable to, and in some cases outperforms, conventional regression baselines, maintaining low error rates in several WP2 scenarios (e.g., Spring-WP2 and Summer-WP2). In contrast, deep learning models demonstrate lower error rates in various WP1 settings; for instance, MLP achieves the best performance in Spring-WP1, while LSTM slightly surpasses Batch-Lasso in Summer-WP1. However, these models may experience considerable degradation in specific modes (e.g., LSTM/MLP in Winter-WP1), indicating that their mode-wise stability depends on the learnability of the particular weather regime and the adequacy of the available training support.

The comprehensive results presented in Table 22 further corroborate a trend consistent with Dataset A: both CNN and MLP achieve superior average accuracy in global metrics, evidenced by higher R^2 values and lower MAE/MSE scores. In contrast, Batch-Lasso exhibits slightly inferior aggregate performance compared to CNN and MLP but significantly surpasses the regression baselines (Lasso, Ridge, OLS, LARS) and demonstrates comparable or superior performance relative to CART overall. Collectively, the findings from Dataset B reinforce previous conclusions: deep learning models generally excel in global average metrics, while Batch-Lasso maintains greater stability across various modes and delivers competitive accuracy with a lightweight footprint. Additionally, Table 23 illustrates that Batch-Lasso necessitates substantially fewer parameters and significantly reduces training and inference time, as well as memory usage, compared to the deep learning baselines and the tree model, thereby underscoring its practicality and engineering advantages for deployment in resource-constrained edge environments.

6. Conclusions

In this paper, we present a multimodal day-ahead short-term photovoltaic (PV) power forecasting framework that integrates data-driven weather partitioning with sparse linear modeling. The meteorological dataset is initially categorized into seasonal groups and subsequently clustered within each season using the CURE algorithm, resulting in representative weather modes for model construction. For each mode, a Batch-Lasso predictor is trained to automatically select the most relevant variables from high-dimensional meteorological inputs while maintaining model interpretability. To solve the Lasso optimization problem efficiently and accurately, we utilize the FISTA algorithm. Furthermore, a batch-style self-updating mechanism with rollback verification is introduced to refresh the forecasting model utilizing newly available data while preventing performance degradation.

Extensive case studies conducted on two photovoltaic (PV) datasets demonstrate the effectiveness of the proposed framework under heterogeneous weather conditions. In comparison to conventional regression baselines, including Lasso, ridge regression, ordinary least squares, and least angle regression, Batch-Lasso consistently achieves better or comparable forecasting performance across various weather modes and on the entire test set. In the mode-wise comparison on Dataset A, Batch-Lasso attains the lowest mean absolute error (MAE) and mean squared error (MSE) in five of the eight weather modes, while maintaining a substantially lighter model structure. External validation on Dataset B reveals a consistent pattern: Batch-Lasso achieves $R^2 = 0.9502$, MAE = 0.835, and MSE = 1.590 on the full test set, clearly outperforming the conventional regression baselines while remaining lightweight with only 44 parameters, 0.10 MB of memory, and a training time of 0.054 s. When compared to representative machine learning baselines such as LSTM, CNN, MLP, and CART, deep models demonstrate superior global average metrics in certain instances. However, the objective of this study is not to achieve universally optimal predictive accuracy across all model configurations. Instead, it aims to develop a lightweight, interpretable, and self-updatable multimode forecasting framework that is practically feasible for deployment on resource-constrained intelligent control terminals. From this perspective, Batch-Lasso presents a competitive trade-off between accuracy and efficiency, along with stable performance across different modes.

Overall, the proposed multimode Batch-Lasso framework presents an interpretable and computationally efficient solution for short-term PV forecasting, thereby facilitating reliable operation and planning in ICT-enabled and data-driven energy systems. However, the current study is constrained to two datasets and short-term validation settings. Despite these limitations, the low model complexity, memory requirements, and training costs associated with the proposed framework render it appealing for sustainable and large-scale deployment on resource-constrained intelligent control terminals. Future work will extend validation to additional sites and climates, incorporate formal distribution-alignment statistics into the update decision, investigate adaptive mode evolution and update scheduling, and conduct longer-horizon online validation to better characterize rollback frequency and model evolution over extended periods.

Author Contributions: Conceptualization, T.Y. and B.C.; methodology, T.Y. and Y.W.; software, B.C. and Y.W.; validation, B.C. and Q.C.; formal analysis, T.Y., D.L.; investigation, Q.C.; resources, B.C. and Y.W.; data curation, B.C.; writing—original draft preparation, T.Y. and B.C.; writing—review and editing, T.Y.; visualization, B.C.; supervision, D.L.; project administration, T.Y. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data presented in this study are available in the reference section. The photovoltaic power output dataset can be found in reference [43,44].

Conflicts of Interest: Author Yuying Wang was employed by National Grid Jiangsu Electric Power Co., Ltd., Xuzhou Jiawang District Power Supply Branch. The remaining authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Appendix A

Table A1. Comparison with representative studies.

Reference	Main Focus	Mode Partitioning	Self-Updating	Lightweight/ Interpretable	Main Difference from This Work
[18]	Similar-day clustering for PV forecasting	Yes	No	Limited	Improves data homogeneity, but does not include model self-updating or sparse regression.
[19]	Multimodal deep clustering	Yes	No	No	Focuses on deep clustering rather than lightweight interpretable forecasting.
[32]	Edge-oriented forecasting trade-off	No	No	Yes	Focuses on complexity–accuracy trade-off, but not on multimode sparse self-updating forecasting.

Table A1. *Cont.*

Reference	Main Focus	Mode Partitioning	Self-Updating	Lightweight/ Interpretable	Main Difference from This Work
[37]	Dynamic forecasting	No	Yes	Limited	Emphasizes dynamic adaptation, but does not integrate explicit multimode partitioning before updating.
[41]	Regression-based PV forecasting	No	No	Yes	Lightweight and interpretable, but does not explicitly address weather-mode heterogeneity or adaptive updating.
Proposed Method	Multimode sparse self-updating forecasting	Yes	Yes	Yes	Integrates weather-mode partitioning, sparse regression, efficient optimization, and rollback-verified self-updating for ICT deployment.

Table A2. Sample sizes of weather modes on Dataset A.

Season	WP1	WP2
Spring	10,570	472
Summer	11,004	114
Autumn	11,195	475
Winter	10,881	634

Table A3. Hyperparameter settings of deep learning baselines.

Model	Key Hyperparameters
LSTM	Hidden units = 200; Number of layers = 1; Sequence length = 1; Learning rate = 0.001; Batch size = 256; Epochs = 100
CNN	Channels = 32, 64; Kernel size = 3; Padding = 1; Dropout = 0.3; Learning rate = 0.001; Batch size = 256; Epochs = 100
MLP	Hidden units = 128, 64; Dropout = 0.3, 0.2; Learning rate = 0.001; Batch size = 256; Epochs = 100

Table A4. Statistical significance tests on Dataset A.

Comparator	DM Statistic	DM p-Value	Wilcoxon p-Value
Lasso	−27.015	<0.001	<0.001
Ridge	−7.478	<0.001	0.011
Classical linear	−7.477	<0.001	0.011
Least angle	−10.420	<0.001	0.002
LSTM	+76.602	1.000	1.000
CNN	+13.596	1.000	1.000
MLP	−2.974	0.002	0.472
CART	+4.331	1.000	0.991

Table A5. Bootstrap 95% confidence intervals of MAE.

Model	MAE	95% CI Lower	95% CI Upper
Batch-Lasso	6.854	6.769	6.942
Lasso	7.502	7.413	7.591
Ridge	7.128	7.037	7.220
Classical linear	7.128	7.037	7.220
Least angle	7.182	7.091	7.273
LSTM	5.445	5.365	5.529
CNN	6.648	6.569	6.729
MLP	7.060	6.982	7.139
CART	6.791	6.681	6.903

https://doi.org/10.3390/su18073319

Table A6. Mode-wise results of deep learning models trained on pooled seasonal data.

Season	Mode	Sample Size	LSTM (R^2/MAE)	CNN (R^2/MAE)	MLP (R^2/MAE)
Spring	WP1	10,570	0.9660/0.03094	0.9672/0.02956	0.9678/0.02802
Spring	WP2	472	0.9558/0.01522	0.9280/0.02047	0.9518/0.01530
Summer	WP1	11,004	0.9843/0.01831	0.9845/0.01886	0.9790/0.02592
Summer	WP2	114	0.9468/0.02063	0.9491/0.02644	0.8654/0.05203
Autumn	WP1	11,195	0.9799/0.02434	0.9791/0.02525	0.9801/0.02432
Autumn	WP2	475	0.9257/0.02737	0.9188/0.03329	0.9181/0.03376
Winter	WP1	10,881	0.9623/0.03017	0.9619/0.03094	0.9612/0.03475
Winter	WP2	634	0.9555/0.02395	0.9580/0.02626	0.9516/0.03127

References

1. National Energy Administration of China. *2025 Renewable Energy Grid-Connected Operation*; National Energy Administration of China: Beijing, China, 2026. Available online: https://www.nea.gov.cn/20260212/742b8c6a078347b0b39de676c05c5d58/c.html (accessed on 19 January 2026).
2. Wang, Y.; Fu, W.; Zhang, X.; Zhen, Z.; Wang, F. Dynamic directed graph convolution network based ultra-short-term forecasting method of distributed photovoltaic power to enhance the resilience and flexibility of distribution network. *IET Gener. Transm. Distrib.* **2023**, *18*, 337–352. [CrossRef]
3. Wei, M.; Wen, M.; Luo, J. A short-term electricity consumption forecasting approach based on feature processing and hybrid modelling. *IET Gener. Transm. Distrib.* **2022**, *16*, 2003–2015. [CrossRef]
4. Perera, M.; De Hoog, J.; Bandara, K.; Senan, D.; Halga, S. Day-ahead regional solar power forecasting with hierarchical temporal convolutional neural networks using historical power generation and weather data. *Appl. Energy* **2024**, *361*, 122971. [CrossRef]
5. Yadav, A.; Khargotra, R.; Lee, D.; Kumar, R.; Singh, T. Novel applications of various neural network models for prediction of photovoltaic system power under outdoor condition of mountainous region. *Sustain. Energy Grids Netw.* **2024**, *38*, 101318. [CrossRef]
6. Wang, X.; Ma, W. A hybrid deep learning model with an optimal strategy based on improved VMD and transformer for short-term photovoltaic power forecasting. *Energy* **2024**, *295*, 131071. [CrossRef]
7. Wang, F.; Lu, X.; Mei, S.; Su, Y.; Zhen, Z.; Zou, Z.; Zhang, X.; Yin, R.; Duić, N.; Shafie-khah, M.; et al. A satellite image data based ultra-short-term solar PV power forecasting method considering cloud information from neighboring plant. *Energy* **2022**, *238*, 121946. [CrossRef]
8. Lu, Z.; Chen, W.; Yan, Q.; Li, X.; Nie, B. Photovoltaic power forecasting approach based on ground-based cloud images in hazy weather. *Sustainability* **2023**, *15*, 16233. [CrossRef]
9. Wu, B.; Hao, J. Enhanced xPatch for short-term photovoltaic power forecasting: Supporting sustainable and resilient energy systems. *Sustainability* **2025**, *17*, 7324. [CrossRef]
10. Liu, R.; Wei, J.; Sun, G.; Muyeen, S.; Lin, S.; Li, F. A short-term probabilistic photovoltaic power prediction method based on feature selection and improved LSTM neural network. *Electr. Power Syst. Res.* **2022**, *210*, 108069. [CrossRef]
11. Yang, J.; He, H.; Zhao, X.; Wang, J.; Yao, T.; Cao, H.; Wan, M. Day-ahead PV power forecasting model based on fine-grained temporal attention and cloud-coverage spatial attention. *IEEE Trans. Sustain. Energy* **2024**, *15*, 1062–1073. [CrossRef]
12. Xiang, X.; Li, X.; Zhang, Y.; Hu, J. A short-term forecasting method for photovoltaic power generation based on the TCN-ECANet-GRU hybrid model. *Sci. Rep.* **2024**, *14*, 6477. [CrossRef] [PubMed]
13. Hu, Z.; Gao, Y.; Ji, S.; Mae, M.; Imaizumi, T. Improved multistep ahead photovoltaic power prediction model based on LSTM and self-attention with weather forecast data. *Appl. Energy* **2024**, *359*, 122709. [CrossRef]
14. Wen, H.; Du, Y.; Chen, X.; Lim, E.; Wen, H.; Yan, K. A regional solar forecasting approach using generative adversarial networks with solar irradiance maps. *Renew. Energy* **2023**, *216*, 119043. [CrossRef]
15. Zhang, Q.; Wang, L.; Hao, Y.; Wan, B.; Che, J.; Guo, H. Ultra-short-term solar irradiance prediction of distributed photovoltaic power stations based on satellite cloud images and clear-sky model. *High Volt. Eng.* **2022**, *48*, 3271–3281. [CrossRef]
16. Si, Z.; Yang, M.; Yu, Y.; Ding, T. Ultra-short-term photovoltaic power prediction method based on satellite image feature region positioning. *High Volt. Eng.* **2021**, *47*, 1214–1223. [CrossRef]
17. Li, H.; Gao, B. Short-term PV power forecasting based on improved VMD and SNS-attention-GRU. *Acta Energiae Solaris Sin.* **2023**, *44*, 1292–1300. [CrossRef]
18. Wu, T.; Hu, R.; Zhu, H.; Jiang, M.; Lv, K.; Dong, Y.; Zhang, D. Combined IXGBoost-KELM short-term photovoltaic power prediction model based on multidimensional similar day clustering and dual decomposition. *Energy* **2024**, *288*, 129770. [CrossRef]
19. Dou, W.; Wang, K.; Shan, S.; Chen, M.; Zhang, K.; Wei, H.; Sreeram, V. A multi-modal deep clustering method for day-ahead solar irradiance forecasting using ground-based cloud imagery and time series data. *Energy* **2025**, *321*, 135285. [CrossRef]

20. Wang, H.; Yan, S.; Ju, D.; Ma, N.; Fang, J.; Wang, S.; Li, H.; Zhang, T.; Xie, Y.; Wang, J. Short-term photovoltaic power forecasting based on a feature rise-dimensional two-layer ensemble learning model. *Sustainability* **2023**, *15*, 15594. [CrossRef]
21. Wu, Z.; Pan, F.; Li, D.; He, H.; Zhang, T.; Yang, S. Prediction of photovoltaic power by the informer model based on convolutional neural network. *Sustainability* **2022**, *14*, 13022. [CrossRef]
22. Javaid, A.; Sajid, M.; Uddin, E.; Wasas, A.; Ayaz, Y. Sustainable urban energy solutions: Forecasting energy production for hybrid solar-wind systems. *Energy Convers. Manag.* **2024**, *302*, 118120. [CrossRef]
23. Wang, S.; Ma, J. A novel GBDT-BiLSTM hybrid model on improving day-ahead photovoltaic prediction. *Sci. Rep.* **2023**, *13*, 15113. [CrossRef]
24. Liang, L.; Su, T.; Gao, Y.; Qin, F.; Pan, M. FCDT-IWBOA-LSSVR: An innovative hybrid machine learning approach for efficient prediction of short-to-mid-term photovoltaic generation. *J. Clean. Prod.* **2023**, *385*, 135716. [CrossRef]
25. Sarmas, E.; Spiliotis, E.; Stama, E.; Marina, V.; Doukas, H. Short-term photovoltaic power forecasting using meta-learning and numerical weather prediction independent long short-term memory models. *Renew. Energy* **2023**, *216*, 118997. [CrossRef]
26. Zhang, X.; Gao, R.; Zhu, C.; Liu, C.; Mei, S. Ultra-short-term prediction of regional photovoltaic power based on dynamic graph convolutional neural network. *Electr. Power Syst. Res.* **2024**, *226*, 109965. [CrossRef]
27. Zhang, C.; Lin, G.; Yu, Y. Short-term PV output interval prediction based on MEEMD-QUATRE-BILSTM. *Acta Energiae Solaris Sin.* **2023**, *44*, 40–54. [CrossRef]
28. Memar, G.; Keynia, F. Solar power generation forecasting by a new hybrid cascaded extreme learning method with maximum relevance interaction gain feature selection. *Energy Convers. Manag.* **2023**, *298*, 117763. [CrossRef]
29. Ouyang, Z.; Li, Z.; Chen, X. Day-ahead photovoltaic power forecasting with multi-source temporal-feature convolutional networks. *Energy Inform.* **2025**, *8*, 68. [CrossRef]
30. Zou, H.; Yang, J.; Yang, C.; Wang, J.; Ma, H.; Zhu, S.; Sun, J. Short-term power prediction of distributed PV based on multi-scale feature fusion with TPE-CBiGRU-SCA. *IET Gener. Transm. Distrib.* **2024**, *18*, 3200–3220. [CrossRef]
31. Gao, X.; Zang, Y.; Ma, Q.; Liu, M.; Cui, Y.; Dang, D. A physics-constrained deep learning framework enhanced with signal decomposition for accurate short-term photovoltaic power generation forecasting. *Energy* **2025**, *326*, 136220. [CrossRef]
32. Shi, J.; Liu, N.; Huang, Y.; Ma, L. An edge computing-oriented net power forecasting for PV-assisted charging station: Model complexity and forecasting accuracy trade-off. *Appl. Energy* **2022**, *310*, 118456. [CrossRef]
33. Ying, C.; Wang, W.; Yu, J.; Li, Q.; Yu, D.; Liu, J. Deep learning for renewable energy forecasting: A taxonomy, and systematic literature review. *J. Clean. Prod.* **2023**, *384*, 135414. [CrossRef]
34. Jurado, M.; Samper, M.; Rosés, R. An improved encoder-decoder-based CNN model for probabilistic short-term load and PV forecasting. *Electr. Power Syst. Res.* **2023**, *217*, 109153. [CrossRef]
35. Casanova, R.; Conde, A. Enhancement of LSTM models based on data pre-processing and optimization of Bayesian hyperparameters for day-ahead photovoltaic generation prediction. *Comput. Electr. Eng.* **2024**, *116*, 109162. [CrossRef]
36. Alcañiz, A.; Grzebyk, D.; Ziar, H.; Isabella, O. Trends and gaps in photovoltaic power forecasting with machine learning. *Energy Rep.* **2023**, *9*, 447–471. [CrossRef]
37. Mohamed, M.; Mahmood, F.; Abd, M.A.; Chandra, A.; Singh, B. Dynamic forecasting of solar energy microgrid systems using feature engineering. *IEEE Trans. Ind. Appl.* **2022**, *58*, 7857–7869. [CrossRef]
38. Yang, T.; Chen, L.; Li, S.; Lv, G.; Xin, Q.; Dong, A. Research on photovoltaic power prediction based on principal component analysis and linear regression. *J. Nanjing Inst. Technol.* **2022**, *20*, 77–83. [CrossRef]
39. Scott, C.; Ahsan, M.; Albarbar, A. Machine learning for forecasting a photovoltaic (PV) generation system. *Energy* **2023**, *278*, 127807. [CrossRef]
40. Guo, Q.; Zhao, C.; Gao, X. An incremental photovoltaic power prediction model considering online updating and catastrophic forgetting. *Sol. Energy* **2025**, *299*, 113787. [CrossRef]
41. Lateko, A.A.H.; Yang, H.-T.; Huang, C.-M. Short-term PV power forecasting using a regression-based ensemble method. *Energies* **2022**, *15*, 4171. [CrossRef]
42. Iheanetu, K.J. Solar photovoltaic power forecasting: A review. *Sustainability* **2022**, *14*, 17005. [CrossRef]
43. Desert Knowledge Australia. Yulara 2 Poly-Si Fixed 2016 Service Station. DKA Solar Centre. Available online: https://dkasolarcentre.com.au/source/yulara/yulara-2-fixed-service-station (accessed on 16 January 2026).
44. Yao, T.; Wang, J.; Wu, H.; Zhang, P.; Li, S.; Wang, Y.; Chi, X.; Shi, M. A photovoltaic power output dataset: Multi-source photovoltaic power output dataset with Python toolkit. *Sol. Energy* **2021**, *230*, 122–130. [CrossRef]