

Article

Forecast-Time-Safe Load Forecasting for Connected and Automated EV Charging Operation: Periodicity-Aware Residual Correction on a Processed Distribution Load Proxy with Public EV Charging Validation

Yaqi Liang

College of Electrical Engineering and New Energy, China Three Gorges University, Yichang 443002, China;
202310150102@ctgu.edu.cn

Abstract

To address the challenge that connected and automated electric vehicle (EV) charging operation requires short-term load forecasts that preserve the current operating level while accurately capturing local ramps and peaks under strict forecast time information constraints, this paper proposes a forecast-time-safe periodicity-aware residual correction (PARC) framework. The primary experiment is a controlled benchmark on a 60-day processed distribution load proxy series, while a charging load series reconstructed from public Boulder, Colorado, EV charging transactions is used as a secondary traceable validation case. Rather than directly predicting the next load value, PARC uses the persistence forecast as the local operating state anchor and learns only the residual correction from admissible lag, rolling statistical, ramp, daily/weekly memory, and cyclic time features. This design enables a controlled comparison between direct load prediction and residual correction under the same feature boundary. In the primary proxy-series setting, PARC-HistGBR achieves a test mean absolute percentage error (MAPE) of 1.527% and a root mean square error (RMSE) of 37.051 kW, outperforming persistence, a validation-selected seasonal blend, same-feature direct tree learners, long short-term memory (LSTM), and bidirectional LSTM (Bi-LSTM). Additional XGBoost, LightGBM, and CatBoost residual variants, together with Seasonal-ETS and SARIMA-daily statistical baselines, support the interpretation that the residual target formulation, rather than one specific learner, accounts for the main gain. Rolling-origin checks, day-block bootstrap intervals, Diebold–Mariano tests, and Wilcoxon signed-rank tests provide supporting evidence within the short-data setting. In the Boulder EV validation case, the model ranking is metric-dependent, with simple persistence remaining strong for percentage metrics and residual/tree models improving selected absolute error metrics. The results indicate that PARC is useful as an auditable forecast-time-safe residual benchmarking framework for connected and automated EV charging operation; they should not be interpreted as evidence of universal superiority on fully traceable EV-rich feeders.



Academic Editor: Grzegorz Sierpiński

Received: 24 May 2026

Revised: 24 June 2026

Accepted: 25 June 2026

Published: 29 June 2026

Copyright: © 2026 by the author.

Published by MDPI on behalf of the
World Electric Vehicle Association.

Licensee MDPI, Basel, Switzerland.

This article is an open access article
distributed under the terms and

conditions of the [Creative Commons](#)

[Attribution \(CC BY\)](#) license.

Keywords: short-term load forecasting; electric vehicle charging load; connected and automated EV charging operation; residual correction; persistence forecast; forecast time safety

1. Introduction

Electric vehicle (EV) charging behavior exhibits pronounced spatiotemporal clustering characteristics and intrinsic stochasticity. When a large number of vehicles connect to the grid within the same time interval, sharp load peaks and steep ramping events can readily occur at the local feeder level, thereby increasing the peak load demand and further compressing the capacity margin of distribution transformers. With the sustained growth in EV penetration reported in recent global market outlooks [1], the impacts induced by such charging behavior have progressively evolved from stochastic disturbances into a persistent operational risk in distribution networks, rendering traditional operation strategies that rely on day-ahead forecasting and offline scheduling increasingly inadequate for responding to system variations in a timely and accurate manner. In this context, minute-level very-short-term load forecasting can provide critical decision support for distribution network congestion management, coordinated EV charging, and the pre-dispatch of demand response resources [2,3].

The key challenge in load forecasting extends beyond merely improving average prediction accuracy. In the context of high EV penetration, the non-stationarity of load profiles is markedly intensified. Recent public charging event and station-level forecasting studies show strong temporal variability and model-dependent forecasting behavior in EV loads [4–6]. Comparative machine learning and deep learning models [7], transfer and meta-learning strategies [8], and hybrid XGBoost–Bi-LSTM structures [9] are all relevant to EV charging load prediction. Consequently, a single-step very-short-term forecasting model should satisfy three essential requirements: preserving the current operating level, responding rapidly to local ramping dynamics, and effectively capturing residual periodic patterns. Even if a model is capable of producing a globally smooth load trajectory, its practical value in operations will be substantially diminished if it fails to accurately characterize peak magnitudes or the timing of ramping occurrences, since such errors directly influence short-term decisions in distribution network operation.

To tackle these challenges, this paper proposes a forecast-time-safe periodicity-aware residual correction framework, denoted as PARC. Instead of directly predicting the next load level, PARC preserves the persistence forecast as the local operating state anchor and learns only the residual correction from admissible periodic, lag, rolling statistical, and ramp features. This shared residual formulation is the core innovation of the paper: it converts very-short-term load forecasting into a constrained correction problem, so that the current level, repeated temporal structure, and local ramp information are jointly exploited without using any information unavailable at the forecast instant. The different PARC variants used in the experiments are consistent instantiations of the same model form; they differ only in the residual learner used inside the framework. The measured EV extension is used to test whether the same forecast-time-safe residual formulation remains competitive on transaction-derived charging load.

1.1. Literature Context and Research Positioning

Existing EV charging load forecasting research has moved from descriptive demand analysis to data-driven prediction at different aggregation levels. Public charging records, station-level load studies, and charging network forecasting experiments show that EV load is shaped by both recurrent temporal regularities and irregular user behavior [4–6]. This means that forecasting accuracy depends not only on the prediction algorithm, but also on what information is available before the forecast is issued.

Machine learning and deep learning models have therefore been widely used for charging station and aggregated EV load prediction. Representative studies include supervised regression for public charging demand [4], deep learning short-term station forecasting [5],

ConvLSTM/BiConvLSTM charging network forecasting [6], comparative machine/deep learning evaluations [7], transfer and meta-learning under limited data [8], and hybrid XGBoost-Bi-LSTM forecasting [9]. These studies show the value of nonlinear models, but their improvements are often associated with richer features, higher model complexity, or different forecasting time scales.

For general short-term load forecasting, recurrent neural networks and residual learning provide two useful reference paradigms. Deep learning models are widely used in this field [2], LSTM has been applied successfully to residential short-term load forecasting [10], and residual learning can focus model capacity on correction components rather than on the full load level [11]. These findings motivate the LSTM, Bi-LSTM, direct learning, and residual learning benchmarks used in this paper.

However, for the forecasting problem oriented towards very-short-term operation of EV-ready distribution networks, when the persistence component already retains the majority of the information about the current load level, it remains necessary to determine whether the next-step load should be obtained through direct prediction or, alternatively, through a learner that operates under the availability constraints of information at the forecast instant and estimates a correction term solely based on periodicity information, lagged terms, and ramping features. Accordingly, this paper proposes PARC as an innovative framework and conducts a comparative analysis against the persistence baseline, the seasonal baseline, direct tree-based models, and the LSTM and Bi-LSTM baseline models. The comparison follows recent forecasting evaluation guidance [2,12], uses time-ordered validation consistent with time-series prediction assessment [13], includes tree-based short-term load-forecasting controls [14], and validates the EV extension on a public charging-transaction dataset. Table 1 summarizes the benchmark role and interpretation of each compared method group.

Table 1. Research positioning of the compared method groups.

Method Group	Benchmark Role	Interpretation for PARC
Persistence	Current-level anchor	Tests whether a learned correction improves the one-step baseline.
Previous-day-naive	Daily cycle baseline	Tests whether daily repetition alone is sufficient.
Direct tree controls	Same features, direct target	Separates the residual target from the tree learner and feature set.
LSTM and Bi-LSTM	Recurrent benchmarks	Checks PARC against nonlinear sequence models under the same data split.
PARC	Residual correction framework	Learns a forecast-time-safe correction to persistence.

1.2. Contributions

This study makes three focused contributions:

- It proposes PARC as a forecast-time-safe residual forecasting model that learns the correction to persistence rather than the load level itself.
- It designs a periodic residual feature layer that integrates short-term lagged terms, same-time-of-day and same-time-of-week memory, rolling statistics, ramping terms, and cyclical time encoding under boundary constraints.
- It reports an EV charging experiment that evaluates transferability on a measured transaction-derived charging load profile and clarifies the metric-dependent boundary of residual correction.

The novelty of this paper is therefore not the isolated use of residual learning, periodic features, or tree ensembles, all of which have precedents in the load-forecasting literature. Rather, the contribution is the explicit combination of (i) a persistence-anchored output decomposition, (ii) a forecast-time-safe feature boundary, (iii) same-feature direct-versus-residual controls, and (iv) an auditable residual correction procedure for one-step EV-ready load forecasting. This positioning weakens the novelty claim from a general residual learning claim to a controlled forecasting problem formulation. Figure 1 summarizes the study route.

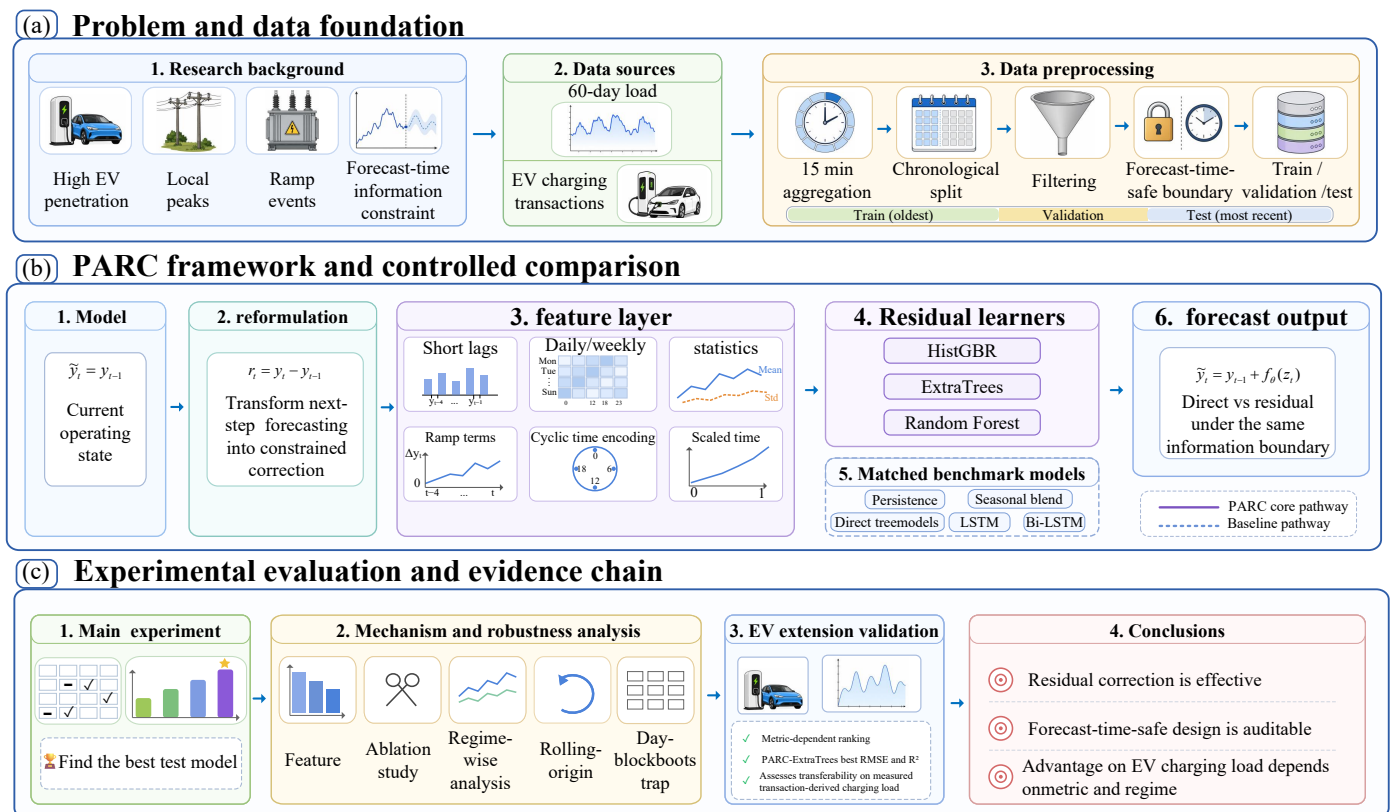


Figure 1. Technical route of this study. The workflow contains (a) problem and data foundation, (b) the PARC framework and controlled comparison, and (c) experimental evaluation and evidence chain.

Figure 1a describes the problem and data foundation. It separates the 60-day processed distribution load proxy series used for controlled method screening from the public Boulder EV charging transaction dataset used for the independently traceable EV extension. This panel also clarifies the preprocessing role of chronological splitting, unit reporting, and forecast-time-safe data preparation. Figure 1b describes the PARC framework and the controlled model comparison. The persistence value is retained as the local operating state anchor; admissible lag, rolling statistical, ramp, and periodic features are constructed without using the target value; and direct and residual learners are compared under the same information boundary. Figure 1c describes the evaluation and evidence chain. It links the main accuracy metrics, regime-wise error analysis, rolling-origin checks, day-block bootstrap intervals, paired statistical tests, computational timing, and the conservative public EV validation to the conclusions reported in the Results and Discussion sections.

2. Materials and Methods

The implementation procedure used in this study is summarized in Figure 2. Unlike Figure 1, which gives the overall study route in the Introduction, this Methods-section

block diagram specifies the executable sequence used in the experiments: data input, chronological preprocessing, forecast-time-safe feature construction, model development, validation-only parameter selection, and final testing with robustness checks. The feedback arrow in the figure indicates that sensitivity results are used to support interpretation rather than to select parameters from the test interval.

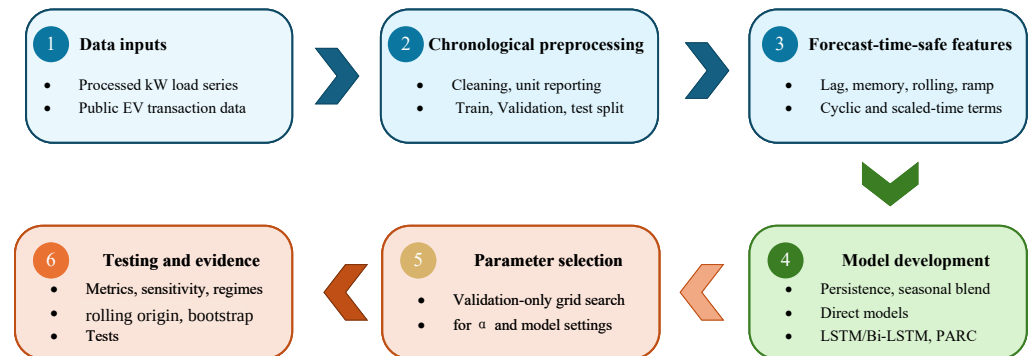


Figure 2. Implementation workflow used in the revised methodology section. The diagram separates validation-only parameter selection from final testing and supporting evidence checks.

2.1. Methodological Overview

Following residual learning ideas for short-term load forecasting [11], the one-step problem is expressed as a persistence-anchored correction. Under very-short-term forecasting horizons, the majority of the information about the next-step load value is already contained in the most recent observation, while the truly difficult part to predict lies in the local correction term. Let the one-step load increment be denoted as:

$$\Delta_t = y_t - y_{t-1}. \quad (1)$$

Then the forecasting problem can be written as

$$y_t = y_{t-1} + \Delta_t, \quad \hat{y}_t = y_{t-1} + \hat{\Delta}_t. \quad (2)$$

Consistent with chronological forecasting evaluation and time-series validation principles [12,13], PARC estimates the conditional correction term through a feature mapping that satisfies the information availability constraint at the forecast instant. The information available prior to the forecast instant can be expressed as

$$\mathcal{I}_t = \{y_s : s \leq t-1\}. \quad (3)$$

The correction model is then defined as

$$\hat{\Delta}_t = f_\theta(z_t), \quad z_t = [\phi_1(\mathcal{I}_t, t), \dots, \phi_p(\mathcal{I}_t, t)]^\top. \quad (4)$$

This expression makes the information boundary explicit: every element of z_t must be computable before the target value is observed, and no feature mapping $\phi_j(\cdot)$ may use any y_s with $s \geq t$. Therefore, the comparison between direct prediction and residual prediction is conducted under identical information availability conditions.

$$\hat{y}_t^d = g_\phi(z_t), \quad \hat{y}_t^r = y_{t-1} + f_\theta(z_t). \quad (5)$$

The corresponding empirical risks on the training interval are

$$\mathcal{L}_d(\phi) = \sum_{t \in \mathcal{T}_t} \ell(y_t, g_\phi(z_t)), \quad \mathcal{L}_r(\theta) = \sum_{t \in \mathcal{T}_t} \ell(y_t - y_{t-1}, f_\theta(z_t)). \quad (6)$$

The direct-versus-residual comparison is therefore not a comparison of different information sets. It is a controlled test of whether changing the prediction target from y_t to $y_t - y_{t-1}$ improves short-horizon forecasting under the same available information.

2.2. Data Sources, Units, and Preprocessing

Two datasets are used, and their roles are intentionally separated. The primary dataset is a 60-day processed distribution load proxy series with 5760 samples at 15 min resolution. The primary proxy series does not include public feeder identification metadata, transformer rating, customer composition, or raw metering channel information. Therefore, it is reported in kW and used as a processed load benchmark for controlled method evaluation rather than as direct evidence of a traceable measured EV-rich feeder. This clarification is important for interpreting MAE, RMSE, ramp magnitude, and axis labels. The processed series has no zero-load intervals and no missing samples after preprocessing. The chronological split is 70% training, 15% validation, and 15% testing, corresponding to 4031, 865, and 864 samples, respectively. The test interval covers nine days. Table 2 summarizes the processed proxy-series statistics used for the controlled benchmark.

Table 2. Statistical characterization of the 60-day processed distribution load proxy series.

Descriptor	Value	Descriptor	Value
Sampling and length	15 min; 5760 samples (60 days)	Reported unit	kW
Minimum/mean/maximum load	1274.202/1843.122/2578.047 kW	Load factor/peak-to-average ratio	0.715/1.399
Mean/95th percentile/maximum 15 min ramp	37.792/92.547/229.324 kW	Weekday/weekend average load	1857.970/1808.478 kW
Low-load samples below 10% of peak	0 samples	Average daily valley/peak time	03:00/17:30

Figure 3 visualizes the processed proxy-series profile, ramp distribution, and daily envelope.

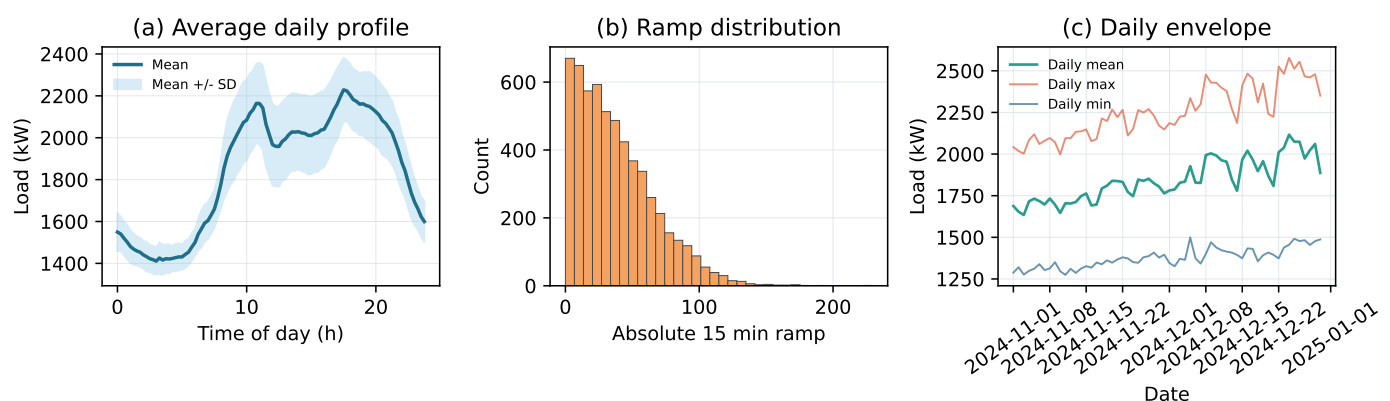


Figure 3. Statistical characterization of the processed distribution load proxy series. (a) Average daily profile with one-standard-deviation band; (b) distribution of absolute 15 min ramps; (c) daily minimum, mean, and maximum load envelope in kW.

The semantic interpretation of the proxy series is a load profile with a pronounced daily cycle, a late-night valley, a morning ramp, and a high-load afternoon/evening plateau. It is therefore suitable for testing whether a short-horizon residual correction can preserve local continuity while responding to ramps. However, because the underlying physical asset and metering context are not publicly traceable, conclusions from this dataset are limited to methodological behavior on this processed load profile. Claims about measured EV charging behavior are instead based on the public Boulder EV charging transaction extension.

For the EV extension, the public Boulder transaction records are cleaned by retaining valid charging sessions, converting start and end times to a common UTC time basis, estimating session average charging power from delivered energy and active charging duration, and aggregating active charging sessions onto a 15 min grid. The resulting series covers 11,520 samples from 1 February 2023 to 1 June 2023, and the load unit is kW. Percentage metrics for this dataset are computed only on positive-load intervals when the denominator would otherwise be zero; MAE, RMSE, WAPE, sMAPE, and R^2 are computed over the full test interval as specified below.

For both datasets, preprocessing is restricted to operations available before the forecasted time. The LSTM and Bi-LSTM inputs are normalized using only the training interval minimum and maximum, and no validation or test observations are used to fit the scaling parameters. For feature-based models, each target row is labeled by the next observed value y_t and by the residual target $r_t = y_t - y_{t-1}$. The feature vector is constructed only from y_{t-1} and earlier observations, time-of-day/time-of-week encodings, and rolling statistics whose windows end at $t - 1$.

2.3. Forecasting Task and Information Boundary

Let $y_t \in \mathbb{R}$ denote the load value in kW at 15 min sampling instant t . For the primary proxy experiment, y_t is the processed distribution load proxy reported in kW; for the Boulder EV experiment, y_t is measured in kW after transaction aggregation. The task is one-step-ahead forecasting: estimate y_t using information available no later than $t - 1$. The forecast horizon is therefore $h = 1$ sample, or 15 min. This boundary is important because any use of y_t , future load values, or normalization fitted on the full series would produce optimistic error estimates. Using the information set in Equation (3), any admissible deterministic forecasting model can be written as a mapping from this information set and the forecast issue time to a scalar prediction:

$$\hat{y}_t = F(\mathcal{I}_t, t; \theta), \quad (7)$$

where θ denotes the trainable parameters or hyperparameters of the forecasting rule. For this study, the mapping is restricted to univariate load history and deterministic point forecasts. Multi-step and probabilistic forecasting are natural extensions, but they require a different evaluation design because recursive forecast errors and forecasted exogenous variables change the information set.

2.4. Benchmark Models

Following recent forecasting benchmark practice [3,12], the benchmark set contains simple baselines, a validation-selected seasonal blend, direct tree controls, recurrent networks, and PARC variants. Let $m = 96$ denote the number of samples in one day and $M = 672$ denote the number of samples in one week. The baselines are:

$$\hat{y}_t^P = y_{t-1}, \quad \hat{y}_t^D = y_{t-m}. \quad (8)$$

A seasonal blend combines immediate persistence and same-time previous-day memory:

$$\hat{y}_t^B = \alpha y_{t-1} + (1 - \alpha) y_{t-m}, \quad \alpha \in [0, 1], \quad (9)$$

with α selected on the validation interval. The weighting parameter controls the balance between the current operating level y_{t-1} and the same-time previous-day memory y_{t-m} . Its inclusion is used only for the seasonal-blend baseline, not for PARC. Because the one-step horizon is 15 min, the current load level is expected to dominate, whereas the previous-day term provides a simple daily-periodic correction. Therefore, α is selected by validation MAPE through a grid search over $\alpha \in \{0, 0.01, \dots, 1\}$, without using any test observations. The validation optimum is $\alpha = 0.87$, which is consistent with a persistence-dominant but not purely persistent baseline. The sensitivity sweep over the same grid shows that values near this region have limited influence on the conclusion: for $\alpha = 0.80, 0.87, 0.91$, and 0.95 , the corresponding test MAPE values are 2.0794%, 2.0096%, 1.9997%, and 2.0168%, respectively. Even the test-best reference value, $\alpha = 0.91$, remains above PARC-HistGBR at 1.5273% MAPE. Thus, the final conclusion is not sensitive to the seasonal-blend weight.

Direct-HistGBR, Direct-ExtraTrees, and Direct-RandomForest predict the load value directly using the same feature vector later used by PARC:

$$\hat{y}_t^G = g_\phi(z_t), \quad \phi^* = \arg \min_{\phi \in \Phi} \sum_{t \in \mathcal{T}_t} \ell(y_t, g_\phi(z_t)). \quad (10)$$

Additional machine learning and statistical controls are included to test whether the residual target remains useful when stronger boosting libraries or traditional seasonal time-series models are added to the benchmark set. Recent load-forecasting studies motivate XGBoost, LightGBM, and CatBoost controls [14–17]; these learners are implemented in both direct and PARC residual forms using the same feature vector and the same chronological information boundary. Two traditional statistical comparators are also included: a seasonal exponential smoothing baseline (Seasonal-ETS) and a daily seasonal ARIMA baseline (SARIMA-daily), following recent forecasting guidance and load-forecasting comparisons that include seasonal ARIMA-type statistical models [12,18]. SARIMA-daily uses an ARMA(1,1) nonseasonal component, a seasonal autoregressive component with the 96-sample daily period, and a constant term. Both statistical models are fitted only on the training interval and are then forecast forward over the validation and test horizons, with the chronological test slice used for reporting. Because the purpose of SARIMA-daily is to provide a reproducible classical statistical reference rather than an exhaustive SARIMA-family model search, its weak result should be interpreted as the performance of this fixed daily seasonal specification on the short proxy series. These models are not used to redefine the proposed framework; they are robustness controls rather than test set tuning parameters.

Following LSTM-based short-term load forecasting benchmarks [10] and EV charging studies using bidirectional recurrent variants [6], the recurrent benchmarks use a one-day historical sequence:

$$x_t = [y_{t-m}, y_{t-m+1}, \dots, y_{t-1}]^T \in \mathbb{R}^m. \quad (11)$$

The LSTM and Bi-LSTM predictors are written abstractly as:

$$\hat{y}_t^L = q_\psi(\text{LSTM}_\psi(x_t)), \quad \hat{y}_t^{BL} = q_\psi(\text{BiLSTM}_\psi(x_t)), \quad (12)$$

where $q_\psi(\cdot)$ is the fully connected regression head. The recurrent network benchmarks are chosen as compact one-day-sequence baselines under validation monitoring rather than as large neural architectures tuned on the test set. For the Bi-LSTM benchmark, a small sensitivity check over hidden size, learning rate, dropout, and depth gives test MAPE

values of 2.290% for one layer with 64 hidden units, 2.227% for one layer with 96 hidden units, 2.342% for a lower-learning-rate/lower-dropout setting, and 2.314% for a deeper two-layer setting. The 96-unit configuration is therefore retained as the Bi-LSTM recurrent baseline. Table 3 gives the implementation settings used to improve reproducibility.

Table 3. Recurrent network benchmark configuration.

Item	LSTM Setting	Bi-LSTM Setting
Input	96 load steps (1 day); calendar only in sensitivity check	96 load steps (1 day)
Architecture	LSTM(64) + dropout 0.20 + FC(32, ReLU) + linear; dense L2	Bi-LSTM(96) + dropout 0.10 + FC/regression output
Optimizer/LR	Adam, 1×10^{-3} , batch 32	Adam mini-batch; 0.001; 0.30 drop after 45 epochs
Training	200 epochs max; early stop 20; restore best validation weights	90 epochs max; validation monitored; gradient threshold 1
Seed/split	Seed 2026; chronological split; no random test sampling	Same protocol

LSTM Benchmark Configuration

The LSTM benchmark was implemented as a direct 15 min ahead forecasting model using the same forecast-time-safe information boundary as the other benchmark models. In the main LSTM and Bi-LSTM benchmarks, the input sequence was constructed from historical load values only within the one-day look-back window, without using calendar/time covariates, the prediction target value, or any information from the test interval during model development. Calendar/time covariates were evaluated separately only in the Calendar-LSTM sensitivity check reported below. The network consisted of one LSTM layer with 64 hidden units, followed by a fully connected layer with 32 neurons and a final linear output neuron for the 15 min ahead load prediction. The LSTM layer used the standard hyperbolic tangent state activation and sigmoid gate activations, while the intermediate dense layer used a ReLU activation function, and the output layer used a linear activation suitable for regression.

To reduce overfitting, dropout with a rate of 0.20 was applied after the LSTM layer, and L2 weight regularization was used in the dense layer. The model was trained using the Adam optimizer with a learning rate of 1×10^{-3} , a batch size of 32, and a maximum of 200 epochs. Early stopping based only on the validation loss was adopted, with a patience of 20 epochs, and the model parameters corresponding to the best validation performance were restored. No test set information was used for architecture selection, hyperparameter tuning, early stopping, or model selection. The LSTM architecture is summarized in Figure 4.

This recurrent comparison is intentionally interpreted with caution. PARC uses engineered periodic memory and rolling/ramp covariates, whereas the main recurrent baselines primarily test whether a compact one-day load-only sequence model is sufficient on this short univariate series. Calendar-LSTM and Calendar-Bi-LSTM are therefore reported as additional sensitivity checks rather than as the main recurrent benchmarks. These calendar feature runs give MAPE values of 1.844% for Calendar-LSTM and 2.478% for Calendar-Bi-LSTM, indicating that adding simple calendar inputs does not overturn the main ranking. Nevertheless, the strongest architectural comparison in this paper is the same-feature direct-versus-residual tree comparison because those models share exactly the same input variables.

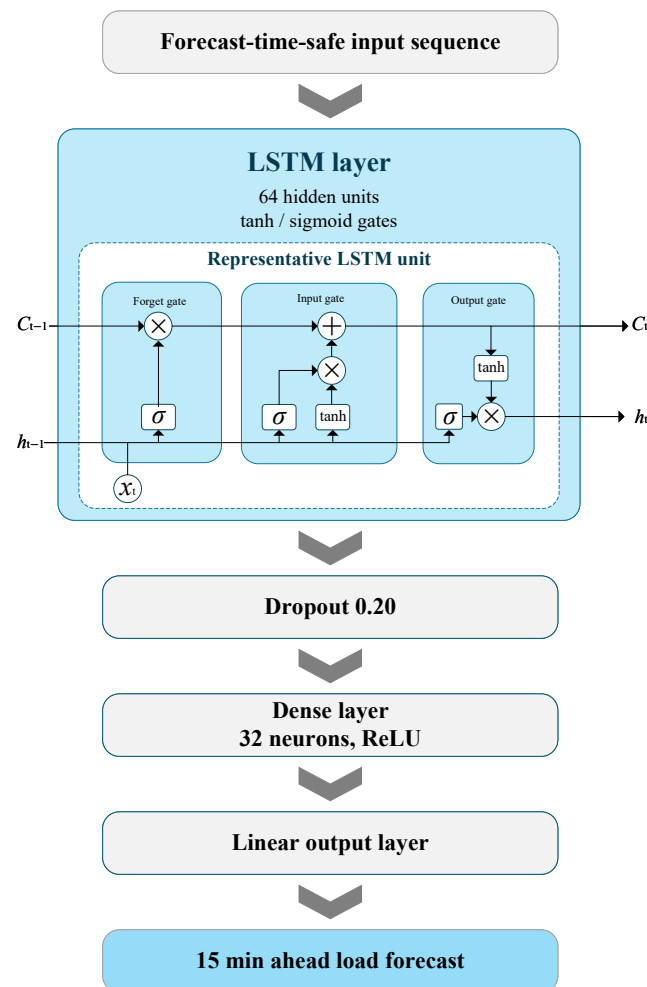


Figure 4. LSTM benchmark architecture used for the recurrent network comparison.

2.5. PARC Framework

As a model-specific adaptation of residual short-term load-forecasting ideas [11], PARC starts from the persistence baseline and learns a residual correction. The baseline, residual target, and final forecast are:

$$\tilde{y}_t = y_{t-1}, \quad (13)$$

$$r_t = y_t - y_{t-1}, \quad (14)$$

$$\hat{y}_t = y_{t-1} + f_\theta(z_t). \quad (15)$$

The residual learner is fitted by empirical risk minimization on the training interval:

$$\theta^* = \arg \min_{\theta \in \Theta} \sum_{t \in \mathcal{T}_t} \ell(r_t, f_\theta(z_t)), \quad (16)$$

where $\ell(\cdot)$ is the regression loss used by the tree learner. This target reformulation reduces the burden on the learner: the model does not need to learn the entire load level, only the correction to a strong local forecast. The feature vector z_t is constructed from information available before t and organized into lag, memory, rolling statistical, ramp, cyclic, and scaled-time components:

$$z_t = [L_t, M_t, S_t, R_t, C_t, \tau_t]. \quad (17)$$

The short-lag and seasonal memory groups are:

$$L_t = [y_{t-a} : a \in \mathcal{A}], \quad \mathcal{A} = \{1, 2, 3, 4, 8, 16, 32\}, \quad (18)$$

$$M_t = [y_{t-m}, y_{t-2m}, y_{t-M}], \quad m = 96, \quad M = 672. \quad (19)$$

The rolling statistical group uses window lengths $\mathcal{K} = \{4, 8, 16, 32, 96\}$, corresponding to 1 h, 2 h, 4 h, 8 h, and 24 h windows at 15 min resolution. These values are selected to cover local fluctuations, half-day operating changes, and daily memory while preserving the forecast time boundary. Sensitivity to the feature window design is examined through the feature-group ablation in the Results section, where the full 41-feature PARC set is compared with short-lag-only and partial memory/rolling variants. For each $k \in \mathcal{K}$:

$$\mu_{t,k} = \frac{1}{k} \sum_{j=1}^k y_{t-j}, \quad \sigma_{t,k} = \sqrt{\frac{1}{k} \sum_{j=1}^k (y_{t-j} - \mu_{t,k})^2}, \quad (20)$$

$$\underline{y}_{t,k} = \min_{1 \leq j \leq k} y_{t-j}, \quad \bar{y}_{t,k} = \max_{1 \leq j \leq k} y_{t-j}, \quad (21)$$

and

$$S_t = [\mu_{t,k}, \sigma_{t,k}, \underline{y}_{t,k}, \bar{y}_{t,k} : k \in \mathcal{K}]. \quad (22)$$

The ramp group contains six changes implemented in the experiment: three short-range differences, one daily persistence-to-memory difference, and two same-time seasonal changes:

$$R_t = [y_{t-1} - y_{t-2}, y_{t-1} - y_{t-4}, y_{t-1} - y_{t-8}, \\ y_{t-1} - y_{t-(m+1)}, y_{t-m} - y_{t-2m}, y_{t-m} - y_{t-M}]. \quad (23)$$

Let $h_t \in \{0, \dots, 95\}$ be the within-day index and $w_t \in \{0, \dots, 671\}$ the within-week index. The cyclic encodings are:

$$s_d(t) = \sin\left(\frac{2\pi h_t}{96}\right), \quad c_d(t) = \cos\left(\frac{2\pi h_t}{96}\right), \quad (24)$$

$$s_w(t) = \sin\left(\frac{2\pi w_t}{672}\right), \quad c_w(t) = \cos\left(\frac{2\pi w_t}{672}\right). \quad (25)$$

The cyclic and scaled-time components are then:

$$C_t = [s_d(t), c_d(t), s_w(t), c_w(t)], \quad \tau_t = \frac{t-1}{n}. \quad (26)$$

Three residual learners are first tested: histogram gradient boosting, extremely randomized trees, and random forest. PARC-HistGBR is specified as the primary proxy-data variant before examining the final test comparison because histogram gradient boosting is a compact boosting-based learner for nonlinear tabular regression. PARC-ExtraTrees and PARC-RandomForest are reported as robustness variants; PARC-ExtraTrees also serves as the validation-selected residual learner. Direct-HistGBR, Direct-ExtraTrees, and Direct-RandomForest use the same feature set but predict y_t rather than r_t , making them direct controls for the residual formulation. Therefore, the main conclusion does not rely on selecting the single best test variant.

The largest lag is 672 samples, the first admissible PARC target is index 673, and the full feature count is 41. This implementation correspondence is important because the benchmark is intended to test a forecast time information boundary rather than post hoc feature construction.

Table 4 gives one fully traceable high-ramp forecasting instance from the test interval. The timestamp is reconstructed on the processed 15 min grid to make the forecast issue time, lagged inputs, residual correction, final forecast, and error calculation inspectable. At this instant, the previous load is 1906.144 kW, the model predicts a positive residual correction of 74.707 kW, and the final forecast becomes 1980.851 kW. The observed value is 2135.468 kW, producing an absolute percentage error of 7.240%. This case is intentionally a difficult high-ramp point rather than an average point.

Table 4. Worked example of one PARC-HistGBR 15 min ahead forecast.

Quantity	Value	Quantity	Value
Forecast timestamp/target index	26 December 2024 08:15/5314	Previous load y_{t-1}	1906.144 kW
Same-time memory y_{t-96}/y_{t-672}	2004.653/1992.839 kW	1 h rolling mean/std.	1811.782/56.657 kW
Recent ramp $y_{t-1} - y_{t-2}$	150.270 kW	Daily phase sine/cosine	0.831/−0.556
Predicted residual $\hat{r}_t$	74.707 kW	Final forecast $y_{t-1} + \hat{r}_t$	1980.851 kW
Observed value/absolute error	2135.468/154.617 kW	Absolute percentage error	7.240%

2.6. Evaluation Metrics and Statistical Checks

Forecast accuracy is measured with MAE, RMSE, MAPE, WAPE, sMAPE, and R^2 . These are standard point forecast accuracy measures used in load-forecasting and general forecasting evaluation [3,12]. MAPE is retained for comparability with the original proxy experiment. In the electric vehicle charging experiment, because the transaction-derived charging load series may contain zero-load or low-load intervals where conventional percentage errors tend to be unstable, WAPE and sMAPE are additionally introduced. For all error metrics, a lower value indicates better performance.

$$\text{MAE} = \frac{1}{n} \sum_i |y_i - \hat{y}_i|, \quad (27)$$

$$\text{RMSE} = \sqrt{\frac{1}{n} \sum_i (y_i - \hat{y}_i)^2}, \quad (28)$$

$$\text{MAPE} = \frac{100}{n} \sum_i \left| \frac{y_i - \hat{y}_i}{y_i} \right|, \quad (29)$$

For the EV charging experiment, let $\mathcal{I}_+ = \{i : y_i > 0\}$ and $n_+ = |\mathcal{I}_+|$. The reported positive-load MAPE is formally denoted by:

$$\text{MAPE}^+ = \frac{100}{n_+} \sum_{i \in \mathcal{I}_+} \left| \frac{y_i - \hat{y}_i}{y_i} \right|. \quad (30)$$

$$\text{WAPE} = 100 \frac{\sum_i |y_i - \hat{y}_i|}{\sum_i |y_i|}, \quad (31)$$

$$\text{sMAPE} = \frac{100}{n} \sum_i \frac{2|y_i - \hat{y}_i|}{|y_i| + |\hat{y}_i|}. \quad (32)$$

For the EV charging experiment, MAPE^+ is computed only on positive actual-load samples because the denominator is otherwise undefined at zero load. WAPE, sMAPE,

MAE, RMSE, and R^2 are computed over the full test interval; sMAPE terms with zero numerator and denominator are omitted.

$$R^2 = 1 - \frac{\sum_i (y_i - \hat{y}_i)^2}{\sum_i (y_i - \bar{y})^2}. \quad (33)$$

Because 15 min errors are serially correlated, individual test points are not treated as independent evidence. A day-block bootstrap, adapted from block bootstrap resampling for dependent time-series observations [19], is therefore used on the nine-day test interval. Blocks contain 96 consecutive samples. For each comparator c , the advantage is:

$$A_c = \text{MAPE}_c - \text{MAPE}_{\text{PARC-HistGBR}}. \quad (34)$$

The empirical confidence interval is obtained from the 2.5th and 97.5th percentiles of the block-resampled distribution:

$$\text{CI}_{95}(A_c) = [Q_{0.025}(A_c^*), Q_{0.975}(A_c^*)]. \quad (35)$$

Because the test interval contains only nine daily blocks, the bootstrap is used as supportive evidence rather than as a strong significance claim. A positive interval indicates that PARC has a lower MAPE under this block resampling design.

A rolling-origin check, consistent with time-series validation practice [13], is also performed for the feature-based models. Four non-overlapping three-day windows are evaluated across the final twelve days. For each window q , the training origin expands up to the start of the test window $\mathcal{W}_q$ and the hyperparameters remain fixed:

$$\text{MAPE}_q = \frac{100}{|\mathcal{W}_q|} \sum_{t \in \mathcal{W}_q} \left| \frac{y_t - \hat{y}_t}{y_t} \right|. \quad (36)$$

Finally, paired statistical tests are used to compare PARC-HistGBR with representative comparators and added controls. For a comparator c , the absolute-percentage-error loss difference is $d_i = \text{APE}_{\text{PARC},i} - \text{APE}_{c,i}$. A one-sided Diebold–Mariano-type forecast accuracy test [20] is applied to test whether the mean loss difference is less than zero, with a Newey–West long-run variance estimate to account for serial correlation. A one-sided Wilcoxon signed-rank test [21] is also applied to the paired absolute error differences as a nonparametric check. These tests are interpreted as supportive evidence because the primary proxy test interval remains short.

3. Results

3.1. Main Test-Interval Accuracy

The main test comparison gives a consistent ordering of methods. PARC-HistGBR has the lowest error in Table 5, with MAE = 29.865 kW, RMSE = 37.051 kW, MAPE = 1.527%, and $R^2 = 0.9895$. Relative to LSTM, the MAPE reduction is $(1.932 - 1.527)/1.932 = 20.96\%$, reported as approximately 21.0%. The reduction relative to Bi-LSTM is larger because Bi-LSTM produces higher peak and amplitude errors in this test interval.

The direct tree controls address the main confounding factor. Direct-HistGBR, Direct-ExtraTrees, and Direct-RandomForest use the same forecast time feature set as PARC but predict the load value directly; their test MAPEs are 2.176%, 2.260%, and 2.163%, respectively. All are less accurate than PARC-HistGBR. The improvement is therefore not explained by the tree learner or feature set alone; the residual target is a major contributor to the observed gain. The validation-selected seasonal blend also remains above PARC, with $\alpha = 0.87$ and test MAPE = 2.010%.

Table 5. Forecast accuracy in the nine-day chronological test interval of the processed kW load series.

Model	MAE (kW)	RMSE (kW)	MAPE (%)	R^2
Persistence	40.436	51.416	2.077	0.9798
Seasonal blend	39.320	50.089	2.010	0.9808
Previous-day-naive	101.219	143.435	4.951	0.8429
Direct-HistGBR	45.308	61.159	2.176	0.9714
Direct-ExtraTrees	47.295	62.433	2.260	0.9702
Direct-RandomForest	44.848	59.236	2.163	0.9732
LSTM	38.648	50.101	1.932	0.9808
Bi-LSTM	44.614	58.338	2.233	0.9740
PARC-HistGBR	29.865	37.051	1.527	0.9895

Figure 5 shows that the numerical advantage in Table 5 is mainly associated with two visible behaviors. First, PARC-HistGBR follows the daily load envelope over the complete test interval without introducing the amplitude inflation observed for Bi-LSTM on several local peaks. Second, the zoomed panel shows that the residual correction is most useful during short ramps and peak shoulders, where a pure persistence forecast is delayed and a direct model can slightly overshoot the load level. The MAPE panel therefore should not be read only as an aggregate ranking; it also indicates that the proposed residual formulation improves the parts of the curve that are most relevant for short-term dispatch, charger scheduling, and reserve preparation. This interpretation is consistent with residual short-term load forecasting, where the learner focuses on the correction around a strong baseline rather than relearning the complete load profile [11], and with the recommendation that forecasting methods be compared on genuine held-out chronological forecasts [12,13].

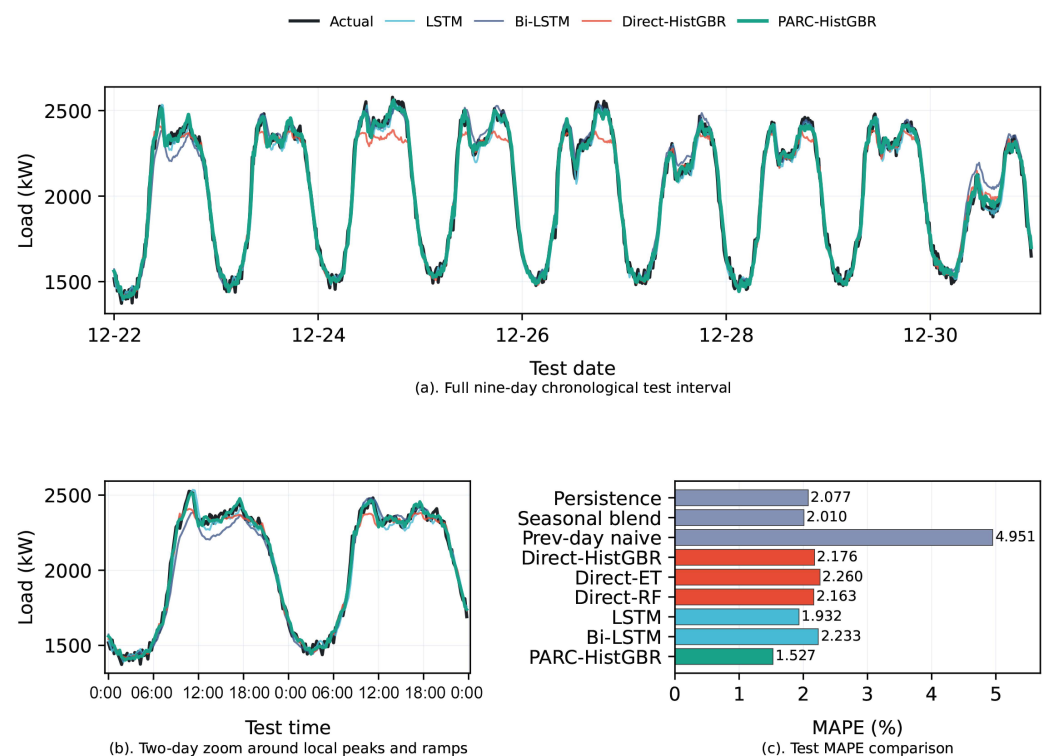


Figure 5. Main performance summary. (a) Full nine-day chronological test interval; (b) two-day zoom around local peaks and ramps; (c) test MAPE comparison across simple baselines, direct tree learners, recurrent networks, and PARC.

The test interval is redrawn in Figure 6 using chronological date labels, daily time-of-day heatmaps, and an error heatmap. The date labels are reconstructed from the processed 15 min grid to make the forecast sequence more interpretable than sample indices alone. The daily heatmap confirms the characteristic low-load early-morning valley, rapid morning ramp, and broad afternoon/evening high-load plateau. The PARC-HistGBR error heatmap shows that the largest errors are localized around abrupt transitions rather than distributed uniformly across all days.

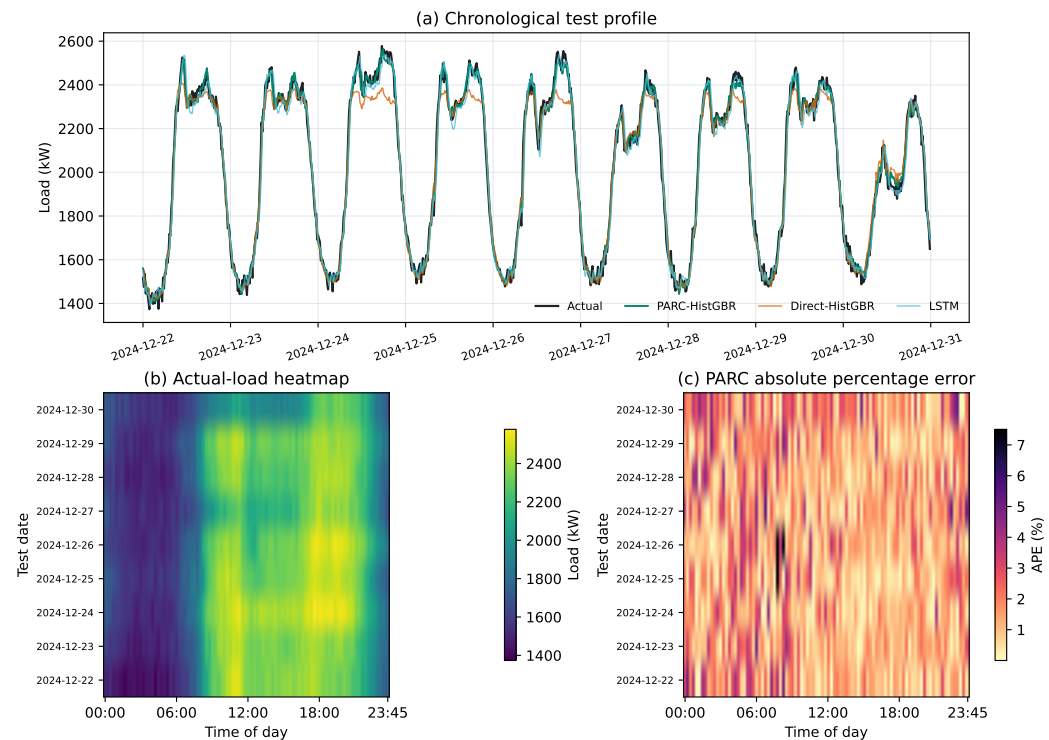


Figure 6. Chronological and diagnostic visualization of the nine-day test interval. (a) Actual load in kW, PARC-HistGBR, Direct-HistGBR, and LSTM forecasts over reconstructed timestamps; (b) actual-load heatmap by test date and time of day; (c) PARC-HistGBR absolute percentage-error heatmap.

Figure 6 further explains why the residual model still produces nonzero errors despite the low average MAPE. The heatmap view shows a stable intra-day structure, but the error panel concentrates the remaining deviations at the morning rise, evening decline, and isolated day-specific changes. These intervals are practically important because they correspond to periods when a charging aggregator or distribution operator would update short-horizon control decisions most frequently. The figure therefore supports a balanced conclusion: PARC improves the average forecast and the main daily shape, but abrupt regime changes remain the dominant source of residual uncertainty.

3.2. PARC Variant Comparison

All residual variants outperform their direct-forecasting counterparts on the main test interval. PARC-ExtraTrees has the lowest validation MAPE, so it can be regarded as the validation-selected residual learner; its test MAPE is 1.544%, close to PARC-HistGBR at 1.527%. This narrow range reduces reliance on a single learner choice and supports the interpretation that the residual formulation and feature boundary are more important than one particular tree ensemble. Table 6 reports the validation and test performance of the direct and residual tree variants.

Table 6. Validation and test the performance of direct and residual tree variants.

Model	Validation MAPE (%)	Test MAPE (%)	Test RMSE (kW)	Test R^2
Direct-HistGBR	1.617	2.176	61.159	0.9714
Direct-ExtraTrees	1.624	2.260	62.433	0.9702
Direct-RandomForest	1.689	2.163	59.236	0.9732
PARC-HistGBR	1.466	1.527	37.051	0.9895
PARC-ExtraTrees	1.426	1.544	37.222	0.9894
PARC-RandomForest	1.452	1.544	37.535	0.9892

Table 7 extends the comparison to additional boosting and statistical baselines.

Table 7. Additional machine learning and statistical baselines with matched PARC residual variants on the processed kW load test interval.

Model	MAE (kW)	RMSE (kW)	MAPE (%)	R^2
Direct-XGBoost	46.657	63.259	2.238	0.9694
PARC-XGBoost	30.285	37.549	1.540	0.9892
Direct-LightGBM	45.132	60.564	2.169	0.9720
PARC-LightGBM	29.886	37.238	1.528	0.9894
Direct-CatBoost	51.740	67.983	2.487	0.9647
PARC-CatBoost	30.388	37.915	1.551	0.9890
PARC-HistGBR	29.865	37.051	1.527	0.9895
Seasonal-ETS	136.822	165.982	6.322	0.7896
SARIMA-daily	371.983	418.213	17.684	−0.3359

The additional baselines strengthen the direct-versus-residual interpretation. Direct-XGBoost, Direct-LightGBM, and Direct-CatBoost remain close to the direct HistGBR controls and do not match the residual variants. PARC-XGBoost, PARC-LightGBM, and PARC-CatBoost, in contrast, approach PARC-HistGBR, with the residual boosting variants lying between 1.527% and 1.551% MAPE. Seasonal-ETS and SARIMA-daily are substantially weaker on this test interval, with MAPE values of 6.322% and 17.684%, respectively. This suggests that fixed daily seasonal smoothing or a fixed daily seasonal ARIMA structure alone is not sufficient to capture the local ramp and peak corrections required by the one-step task. The SARIMA-daily result should therefore be read as the outcome of a fixed, reproducible daily seasonal specification rather than as a fully optimized SARIMA-family search. Overall, the main contribution is the residual-periodic formulation under the forecast time information boundary rather than the exclusive use of one boosting implementation.

The feature-based machine learning models and Seasonal-ETS in Table 8 require less than four seconds for fitting and less than 0.03 ms per one-step test forecast on the local workstation used for the experiment. SARIMA-daily is much more expensive to fit because the seasonal state space model includes the 96-sample daily period, although its test time forecast generation remains below 1 ms per sample. The reported timings are implementation-dependent, but they indicate that the proposed residual correction design and the added boosting controls are computationally compatible with 15 min operational forecasting, while SARIMA-daily is included mainly as a traditional statistical benchmark.

Table 8. Computational timing of additional feature-based and statistical models on the processed kW-load experiment.

Model	Training Time (s)	Total Test Inference (ms)	Inference per Sample (ms)
Direct-XGBoost	0.775	6.047	0.0070
PARC-XGBoost	0.591	6.453	0.0075
Direct-LightGBM	0.657	11.448	0.0132

Table 8. Cont.

Model	Training Time (s)	Total Test Inference (ms)	Inference per Sample (ms)
PARC-LightGBM	0.613	12.452	0.0144
Direct-CatBoost	1.862	9.008	0.0104
PARC-CatBoost	1.586	3.303	0.0038
PARC-HistGBR	3.505	19.124	0.0221
Seasonal-ETS	0.578	11.567	0.0134
SARIMA-daily	331.716	451.278	0.5223

3.3. Feature Importance Evaluation and Ablation Analysis

The leading features are daily cyclic encodings and recent ramp terms. The most important features are $\cos_{\text{day}}$, diff_1 , $\sin_{\text{day}}$, and diff_4 , followed by lag_{16} and diff_{day} . This pattern is mechanistically consistent with PARC: the forecast preserves local continuity through persistence and then adjusts that baseline according to daily phase, recent direction, and volatility. Table 9 lists the top twelve PARC features from the ExtraTrees residual learner.

Table 9. Top twelve PARC features from the ExtraTrees residual learner.

Rank	Feature	%	Rank	Feature	%	Rank	Feature	%
1	$\cos_{\text{day}}$	23.869	5	lag_{16}	6.619	9	mean_{32}	4.501
2	diff_1	16.372	6	diff_{day}	6.195	10	lag_1	4.464
3	$\sin_{\text{day}}$	11.945	7	diff_8	5.051	11	std_4	4.099
4	diff_4	8.205	8	std_{16}	4.665	12	std_{32}	4.014

Table 10 reports the PARC feature-group ablation results.

Table 10. Ablation study of PARC feature groups.

Variant	Features	MAE	RMSE	MAPE (%)	R^2
Persistence only	0	40.436	51.416	2.077	0.9798
Residual short-lag	10	34.563	43.214	1.734	0.9857
Short-lag + memory	16	36.762	46.476	1.854	0.9835
Memory + rolling statistics	36	34.873	43.820	1.768	0.9853
Full PARC feature set	41	29.865	37.051	1.527	0.9895

Memory denotes the added daily and weekly lag groups. Persistence alone gives 2.077% MAPE. Residual short-lag features reduce MAPE to 1.734%, showing that recent local changes contain useful correction information. The full PARC feature set reduces MAPE to 1.527%. Intermediate variants are not strictly monotonic, which is expected in short time series; nevertheless, the full forecast-time-safe residual feature set gives the best test accuracy.

Figure 7 connects the model mechanism to the observed forecasting behavior. The importance ranking emphasizes recent lagged changes and cyclic time descriptors, which means that the residual learner mainly uses local ramp information together with recurring daily timing. The ablation panel gives the same message from the opposite direction: short-lag residual features already improve on persistence, while the complete set of lags, rolling statistics, ramp descriptors, and calendar encodings gives the lowest error. The non-monotonic intermediate variants show that simply adding one group of features is not always sufficient on a short series; the strongest result comes from combining local dynamics and periodic context. This is consistent with EV and load-forecasting studies that report strong daily regularity but also substantial short-term variability in charging demand [4–6].

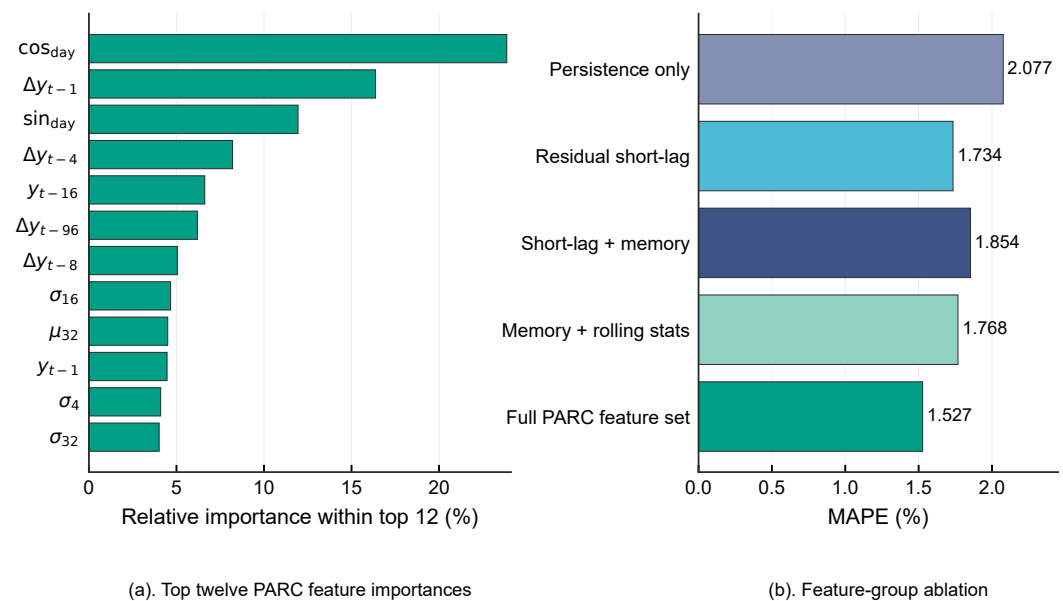


Figure 7. Feature mechanism summary. (a) Top twelve PARC feature importances from the ExtraTrees residual learner; (b) feature-group ablation, showing the incremental value of the full forecast-time-safe feature set.

3.4. Regime-Wise Error Analysis

PARC-HistGBR achieves the lowest MAPE in the valley, middle-load, peak-load, and high-ramp subsets. Among them, the high-ramp subset represents the most operationally revealing condition, as ramp errors are more likely to affect feeder and transformer capacity margins. In this case, PARC reduces the MAPE from 3.597% for LSTM and 2.973% for Bi-LSTM to 2.298%. Table 11 reports these regime-wise MAPE comparisons.

Table 11. MAPE comparison across different operating conditions.

Regime	Persistence	LSTM	Bi-LSTM	PARC-HistGBR
Valley (bottom 25%)	2.370	1.907	2.005	1.839
Middle (25–75%)	2.309	2.165	2.645	1.577
Peak (top 25%)	1.318	1.493	1.636	1.116
High-ramp (top 25%)	4.401	3.597	2.973	2.298

3.5. Rolling-Origin and Block Bootstrap Checks

Across four rolling-origin windows, PARC-HistGBR has the lowest mean MAPE at 1.497% with a standard deviation of 0.110 percentage points. The direct tree models range from 1.691% to 1.738% mean MAPE, again supporting the residual target. Table 12 summarizes the rolling-origin MAPE statistics.

Table 12. Rolling-origin MAPE over four three-day windows.

Model	Mean $\pm$ SD (%)	Min. (%)	Max. (%)
Persistence	2.055 $\pm$ 0.052	1.989	2.098
Seasonal blend	2.035 $\pm$ 0.135	1.899	2.167
Direct-HistGBR	1.691 $\pm$ 0.230	1.450	1.959
Direct-ExtraTrees	1.718 $\pm$ 0.196	1.470	1.946
Direct-RandomForest	1.738 $\pm$ 0.206	1.496	1.999
PARC-HistGBR	1.497 $\pm$ 0.110	1.426	1.660

Table 13 reports the day-block bootstrap MAPE advantage intervals.

Table 13. Day-block bootstrap MAPE advantage of PARC-HistGBR.

Comparator	Advantage (pp)	95% CI
Persistence	0.549	[0.469, 0.630]
Seasonal blend	0.482	[0.370, 0.638]
Direct-HistGBR	0.649	[0.335, 1.049]
Direct-ExtraTrees	0.733	[0.402, 1.168]
Direct-RandomForest	0.636	[0.342, 1.029]
LSTM	0.405	[0.293, 0.472]
Bi-LSTM	0.705	[0.297, 1.193]

The block bootstrap intervals remain positive for all reported comparators. Relative to LSTM, the MAPE advantage is 0.405 percentage points with a 95% interval of [0.293, 0.472]. Relative to Direct-HistGBR, the advantage is 0.649 percentage points with a 95% interval of [0.335, 1.049].

Table 14 provides formal paired-test support for lower paired APE losses under the specified nine-day test design. The nonsignificant rows for PARC-XGBoost, PARC-LightGBM, and PARC-CatBoost show that several residual boosting variants are statistically close to PARC-HistGBR in this short test interval, which again supports interpreting PARC as a residual correction framework rather than as a claim about a unique boosting learner.

Table 14. Paired statistical tests comparing PARC-HistGBR with representative comparators and added controls on the nine-day test interval.

Comparator	Mean APE Advantage (pp)	DM p on APE Loss	Wilcoxon p
Persistence	0.549	2.57×10^{-22}	4.33×10^{-23}
Seasonal blend	0.482	6.20×10^{-17}	2.52×10^{-21}
Direct-HistGBR	0.649	3.62×10^{-7}	6.65×10^{-25}
Direct-ExtraTrees	0.733	6.47×10^{-9}	2.79×10^{-35}
Direct-RandomForest	0.636	1.74×10^{-8}	2.69×10^{-28}
LSTM	0.405	7.05×10^{-14}	1.99×10^{-13}
Bi-LSTM	0.705	5.45×10^{-9}	7.46×10^{-28}
Direct-CatBoost	0.960	3.45×10^{-11}	2.05×10^{-42}
PARC-XGBoost	0.013	0.181	0.121
PARC-LightGBM	0.001	0.467	0.309
PARC-CatBoost	0.023	0.165	0.236
Seasonal-ETS	4.795	2.16×10^{-36}	1.78×10^{-117}
SARIMA-daily	16.156	5.28×10^{-116}	9.40×10^{-142}

Figure 8 shows that the improvement is not confined to one favorable test summary. In the regime-wise panel, PARC-HistGBR remains competitive in valley, ramp, and high-load conditions, which is important because these regimes stress different forecasting mechanisms: valleys test low-load stability, ramps test fast adaptation, and high-load periods test peak tracking. The rolling-origin panel shows that the ranking is repeated across short chronological windows rather than being driven by a single day. The bootstrap panel then summarizes the same evidence at the daily-block level, where the positive advantage values indicate that PARC-HistGBR generally lowers MAPE relative to the direct, recurrent, and naive comparators. The practical implication is that the residual correction is useful across operating conditions, although the short test horizon means the result should be interpreted as evidence of robustness within this dataset rather than as a universal performance guarantee.

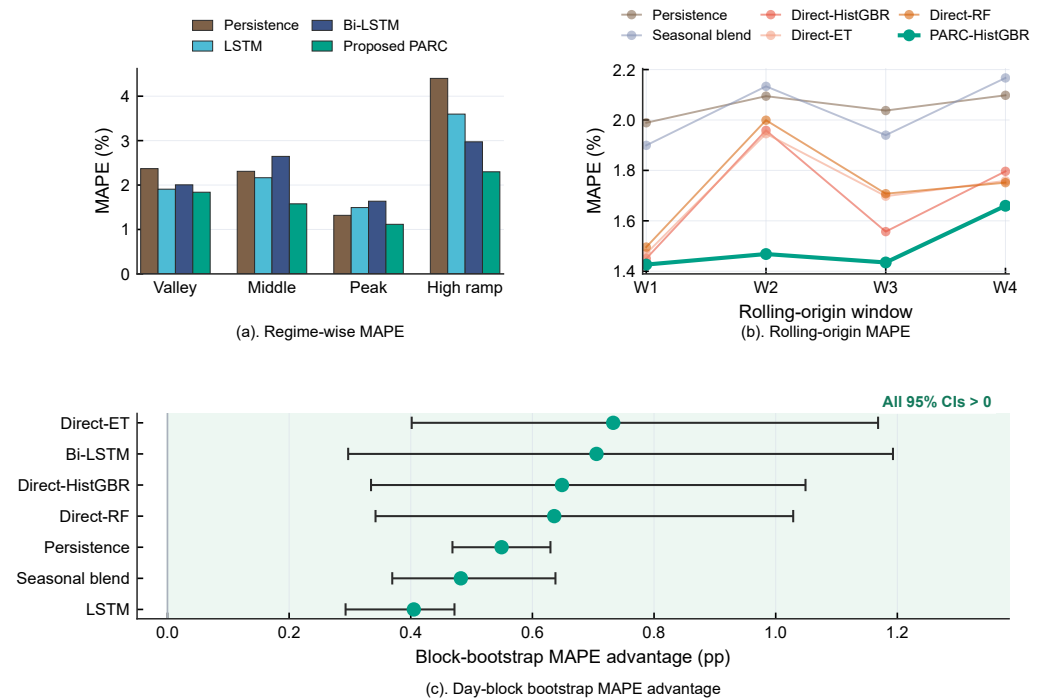


Figure 8. Robustness summary. (a) Regime-wise MAPE for representative models; (b) rolling-origin MAPE across four three-day windows; (c) day-block bootstrap MAPE advantage of PARC-HistGBR relative to the selected comparators shown, where positive values indicate lower MAPE for PARC-HistGBR.

3.6. Validation on Public EV Charging Transaction

In the following, the electric vehicle charging experiment is conducted. The Boulder dataset is derived from EV charging transaction records and has been converted into an aggregated charging load profile at a 15 min granularity.

The EV-specific experiment supports a conservative interpretation of the proposed framework. The ranking is mixed across metrics: persistence and the seasonal blend give the lowest sMAPE and positive-load MAPE, Direct-HistGBR gives the lowest MAE and WAPE, and PARC-ExtraTrees gives the lowest RMSE and highest R^2 . The low positive-load MAPE and sMAPE of persistence indicate that many positive-load intervals are locally continuous; however, tree-based models reduce absolute error metrics such as MAE, WAPE, and RMSE. Therefore, the EV experiment supports the usefulness of forecast-time-safe periodic features.

Figure 9 clarifies why the EV extension gives a more mixed ranking than the smoother proxy-load experiment. The chronological panel shows long locally continuous segments separated by sharp charging changes; in such intervals, persistence can obtain a low percentage error when the load stays positive and changes slowly. The two-day zoom, however, shows that tree-based models better reduce absolute deviations around larger charging pulses. This explains the metric-dependent result in Table 15: percentage metrics reward local continuity, whereas MAE, WAPE, RMSE, and R^2 are more sensitive to the magnitude of missed charging events. Similar intermittency and time-of-day dependence have been reported in EV charging forecasting and charging-demand studies [4,5,7].

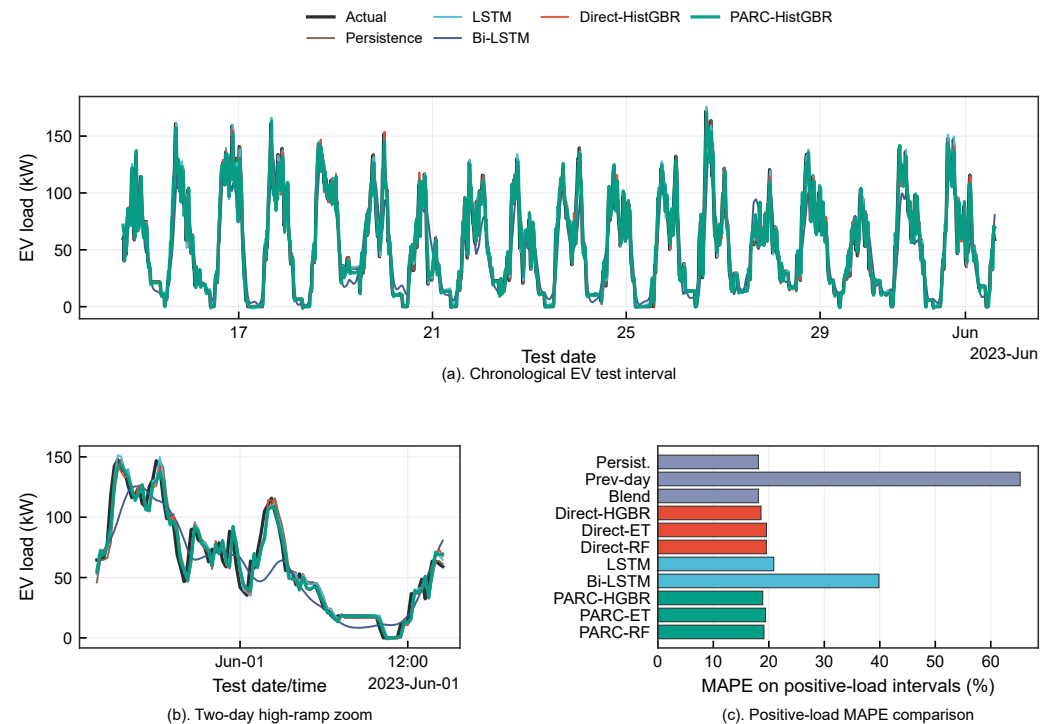


Figure 9. Public EV charging data validation summary. (a) Chronological EV test interval; (b) two-day high-ramp zoom; (c) positive-load MAPE comparison for persistence, recurrent, direct model, and PARC variants.

Table 15. Validation on the public Boulder EV charging dataset. MAPE is computed over positive actual-load intervals; the other metrics are computed over the full test interval. Best values are shown in bold; RF denotes random forest.

Model	MAE (kW)	RMSE (kW)	WAPE (%)	sMAPE (%)	MAPE ⁺ (%)	R ²
Persistence	6.332	9.529	11.798	16.670	18.114	0.9462
Previous-day naive	21.664	29.422	40.364	65.659	65.296	0.4868
Seasonal blend	6.332	9.529	11.798	16.670	18.114	0.9462
LSTM	6.143	8.805	11.445	30.542	20.900	0.9540
Bi-LSTM	12.903	17.607	24.042	43.937	39.852	0.8162
Direct-HistGBR	5.947	8.661	11.080	30.073	18.590	0.9555
Direct-ExtraTrees	6.114	8.819	11.392	30.648	19.572	0.9539
Direct-RF	6.144	8.798	11.447	30.758	19.554	0.9541
PARC-HistGBR	5.962	8.704	11.108	30.226	18.903	0.9551
PARC-ExtraTrees	5.956	8.610	11.097	30.164	19.416	0.9561
PARC-RF	6.059	8.651	11.289	30.374	19.123	0.9556

Figure 10 provides a daily-profile view of the same EV behavior. Unlike the proxy series, the public charging load contains narrow peaks produced by individual or clustered charging sessions. PARC-HistGBR tracks the broad timing and magnitude of repeated daily activity, but the wireframe cannot exactly reproduce every isolated spike because those spikes depend on stochastic user arrival and session duration. This visual evidence supports the practical interpretation that the proposed framework is better suited to short-horizon operational adjustment than to perfectly predicting every individual charging event.

Figure 11 separates the magnitude and location of the remaining EV errors. The absolute error surface shows that errors are concentrated around high-power charging pulses rather than during the low-load background periods. The PARC-minus-direct difference surface also shows that the residual formulation is not uniformly superior at

every EV interval; it reduces error in some pulse and transition regions but can be slightly worse where the direct model happens to match an isolated event. This explains why PARC-ExtraTrees has the best RMSE and R^2 , while Direct-HistGBR has the smallest MAE and WAPE in Table 15.

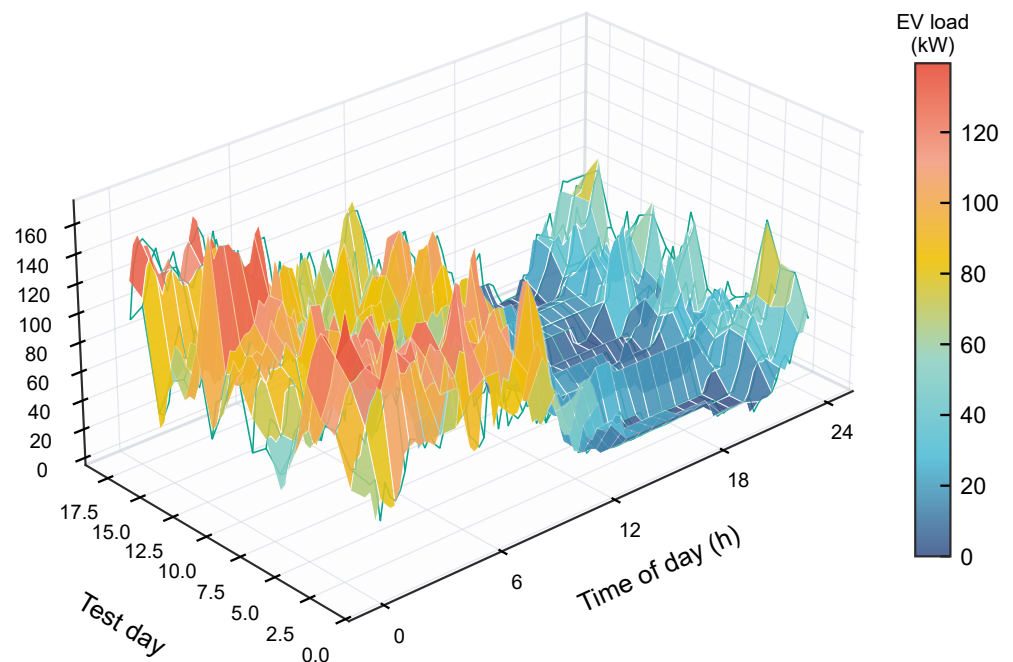


Figure 10. Three-dimensional daily-profile visualization for the public EV charging extension. The measured transaction-derived EV load is shown as a daily surface, and the PARC-HistGBR forecast is overlaid as a green wireframe. The figure highlights the intermittency and sharp daytime charging peaks that distinguish the EV extension from the smoother distribution load proxy.

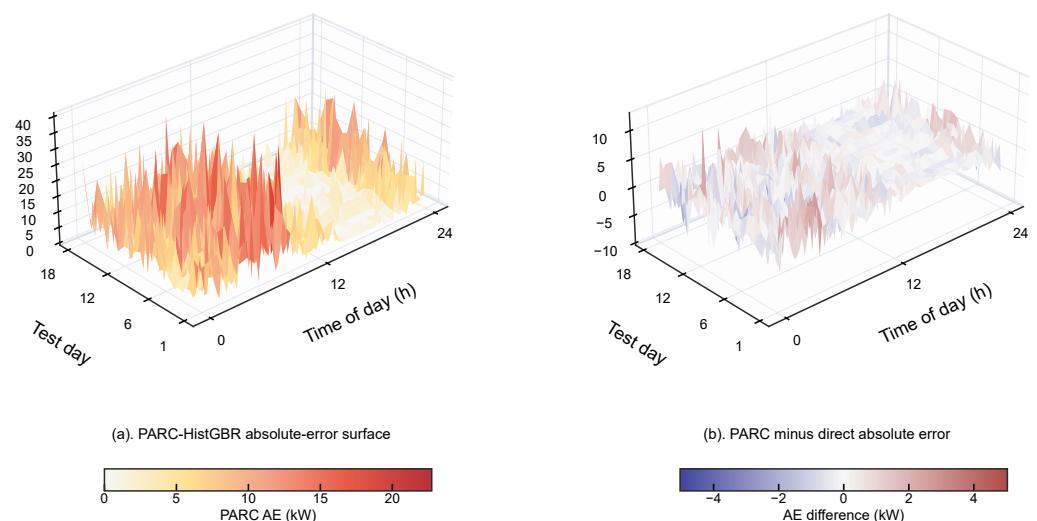


Figure 11. Three-dimensional EV error diagnostics. (a) PARC-HistGBR absolute error surface in kW; (b) difference between PARC-HistGBR and Direct-HistGBR absolute error surfaces. Positive values indicate intervals where PARC-HistGBR has larger absolute error than the matched direct HistGBR model, and negative values indicate intervals where PARC-HistGBR has smaller absolute error.

Figure 12 makes the residual learning task explicit. The actual persistence residual has alternating positive and negative ridges when charging starts, increases, ends, or drops, while the learned correction captures the smoother and more repeatable part of that structure. The remaining mismatch is expected because a 15 min aggregate EV series

contains user-driven events that are only partly predictable from past load and time covariates. Together with Figures 9–11, this figure supports a cautious comparison with existing EV forecasting work: forecast-time-safe historical and cyclic information can improve operational metrics, but the stochastic component of public charging demand limits point forecast accuracy [4–6].

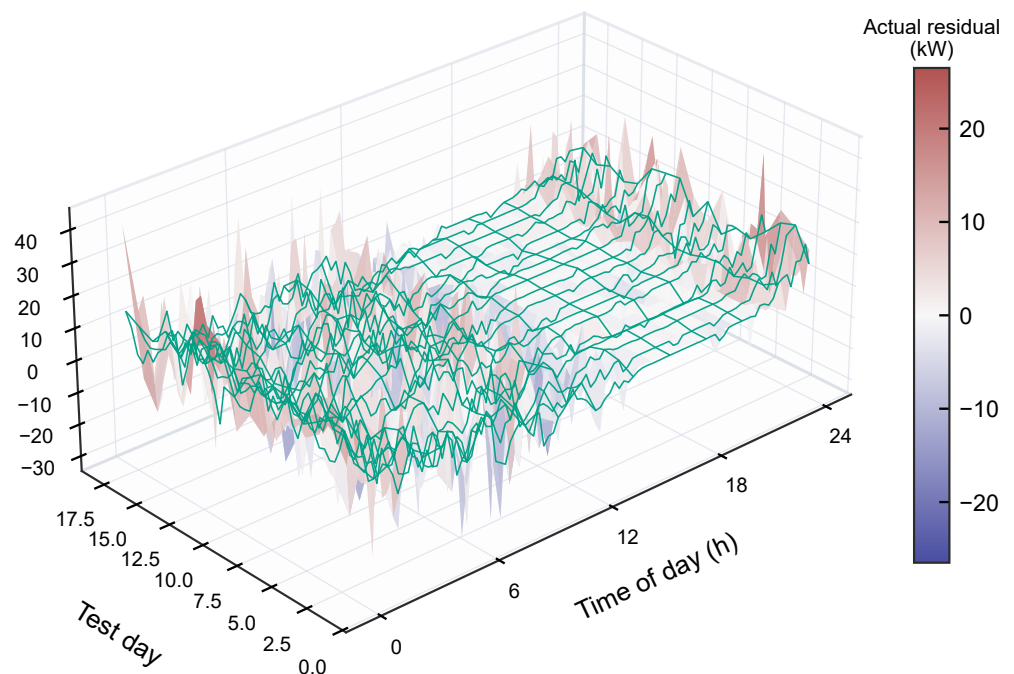


Figure 12. Three-dimensional residual correction diagnostic for the public EV charging extension. The surface shows the actual persistence residual, and the green wireframe shows the learned PARC-HistGBR correction. The comparison makes the residual target formulation visually inspectable and explains how PARC adjusts the persistence baseline on measured EV charging data.

3.7. Last-Day Case with the Original Short Lookback

The last-day case covers a single operating day and employs a shorter input window. Under this setting, Bi-LSTM improves upon the persistence baseline but still does not outperform LSTM. This case is retained only as a legacy comparison with the original short-lookback recurrent experiment; the main evidence for PARC comes from the longer chronological benchmark and the measured EV extension. Table 16 reports the last-day rolling forecast accuracy for the 8-step lookback setting.

Table 16. Last-day rolling forecast accuracy with an 8-step lookback. Best values are shown in bold.

Model	MAE	RMSE	MAPE (%)	R^2
Persistence	42.759	52.451	2.310	0.9571
Previous-day naive	195.336	241.314	10.168	0.0923
LSTM	36.012	45.552	1.921	0.9677
Bi-LSTM	38.938	49.651	2.068	0.9616

4. Discussion

4.1. Interpretation of the Main Result

First, PARC-HistGBR achieves the lowest error on the held-out chronologically ordered test interval. Second, direct tree learners using the same feature set remain consistently less accurate than their residual counterparts. Third, both the rolling-origin validation

and the day-block bootstrap check preserve the direction of the effect across contiguous temporal blocks. Figure 13 visualizes the last-day forecasts under the legacy 8-step lookback setting.

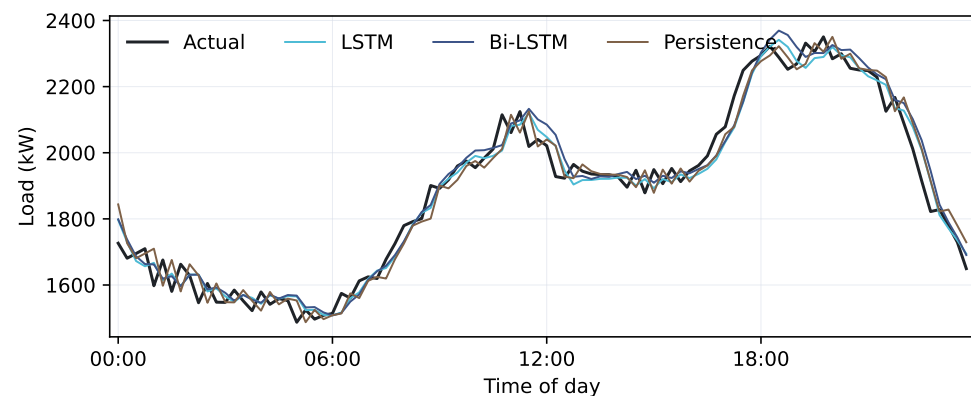


Figure 13. Last-day forecasts using the 8-step lookback setting.

Under this controlled setting, it is the construction of the target variable - rather than the complexity of the recurrent architecture - that accounts for the majority of the observed performance improvement. The persistence component already captures the current operating level, while PARC further introduces a data-driven residual correction guided by intra-day phase, recent ramp direction, rolling volatility, and same-time memory. This design explains why the method performs well on the high-ramp and peak-load subsets.

The comparison with direct tree models is central to this interpretation. If the direct tree learners had been able to match PARC in the proxy experiment, the performance gain could have been attributed solely to the learner family and the feature set. However, all direct tree variants prove less accurate than PARC. This contrast supports the view that reformulating the target as a residual constitutes the core mechanism of the proposed model.

The EV charging experiment provides measured data support for the model while softening the conclusion boundary. On the transaction-derived Boulder charging load profile, persistence and the seasonal blend have the lowest sMAPE and positive-load MAPE, Direct-HistGBR has the lowest MAE and WAPE, and PARC-ExtraTrees has the lowest RMSE and highest R^2 . This indicates that PARC-type residual learners remain competitive on measured EV charging data, but the EV experiment does not demonstrate metric-independent superiority. Accordingly, the EV extension is used as a secondary validation of transferability to traceable charging data rather than as the source of the primary performance claim.

4.2. EV-Ready Operation Scope

These results are relevant to the forecasting layer used in EV-ready distribution-network operation, because the mechanism under investigation targets difficulties that also emerge in charging-rich feeders: short-horizon continuity, daily periodic repetition, rapid variations, and peak tracking. However, the primary evidence remains a processed proxy-series benchmark and should not be read as proof of universal performance on all connected and automated EV charging deployments.

In practical charging station deployments, the PARC feature vector can be further extended to incorporate features such as charger occupancy status, plug-in counts, charging session duration, day type, electricity price periods, and weather forecasts. The same information-boundary rule remains applicable: a feature is valid only if it is known before the forecast horizon.

The present study is a forecasting study rather than a bidirectional charging dispatch study. Therefore, it does not model charger-level power limits, vehicle-to-grid or vehicle-to-home exchange, battery degradation, battery lifetime cost, state-of-charge constraints, or user mobility constraints. These quantities would be required for an optimization or control paper on EV charging/discharging operation, but they are outside the deterministic one-step load-forecasting boundary examined here.

4.3. Limitations and Future Work

This study has five main limitations. First, the empirical scope is limited to the available processed kW load series and the selected public EV charging dataset; broader seasonal, multi-site, and multi-regime generalization still requires additional traceable sites and longer observation periods. The nine-day primary test interval and the limited number of daily bootstrap blocks therefore provide supporting diagnostic evidence rather than strong operational generalization. Second, the primary experiment is univariate; weather, tariff, calendar, and charging session features are not included in the main controlled error comparison. Third, the recurrent benchmarks use a one-day sequence input, whereas PARC explicitly incorporates weekly same-time memory; future work should include recurrent models with longer lookback windows, richer exogenous covariates, or explicitly seasonal recurrent architectures beyond the simple Calendar-LSTM sensitivity checks. Fourth, the EV conclusion is based on deterministic point forecast metrics and mean rank comparisons among the tested models; application-specific cost functions may assign different weights to RMSE, MAPE, WAPE, or ramp errors. Fifth, only one-step-ahead deterministic point forecasting is evaluated, so the method does not provide prediction intervals, quantiles, or scenario trajectories. Probabilistic PARC variants would be necessary for risk-aware congestion management, reserve scheduling, or chance-constrained EV operation.

Future work should evaluate the same residual-periodic framework on measured charging station or charging-rich feeder data, include exogenous variables available before the forecast horizon, test multiple seasons, and extend the design to multi-step probabilistic forecasting. Additional comparisons with support vector regression, temporal convolutional networks, and transformer-based models would further strengthen the benchmark.

5. Conclusions

This study proposes a forecast-time-safe periodicity-aware residual correction framework for one-step-ahead short-term load forecasting and evaluates it under chronological validation. The method learns the correction to a persistence forecast using only admissible lag, rolling, ramp, memory, and cyclic features. This formulation directly targets the residual error that remains after local continuity is accounted for.

In the main processed proxy test interval, PARC-HistGBR achieved MAPE = 1.527%, RMSE = 37.051 kW, and $R^2 = 0.9895$, outperforming the persistence baseline, the validation-selected seasonal blend, the direct tree learners, LSTM, and Bi-LSTM. Additional XGBoost, LightGBM, and CatBoost residual variants, together with Seasonal-ETS and SARIMA-daily statistical baselines, show the same pattern: direct boosting and fixed seasonal statistical models remain weaker, whereas residual boosting variants approach PARC-HistGBR. The direct tree controls demonstrate that using the same features to predict load directly is not sufficient in this experiment; the residual prediction target is a major contributor to the performance improvement. The day-block bootstrap intervals, rolling-origin windows, and paired statistical tests provide additional supporting evidence while preserving the limited empirical scope and positioning the method as an auditable residual benchmarking layer for connected and automated EV charging management.

The public Boulder EV charging extension provides measured EV charging evidence but also clarifies the boundary of the claim. On the transaction-derived EV load series, the results are metric-dependent: persistence and the seasonal blend have the lowest sMAPE = 16.670% and positive-load MAPE = 18.114%, Direct-HistGBR has the lowest MAE = 5.947 kW and WAPE = 11.080%, and PARC-ExtraTrees gives the lowest RMSE = 8.610 kW and the highest $R^2 = 0.9561$. The LSTM benchmark remains competitive but does not dominate the best feature-based models, and Bi-LSTM is weaker in this EV extension. Therefore, PARC should be interpreted as a transparent residual benchmarking strategy rather than as a metric-independent best forecaster.

Overall, the contribution of this paper lies in specifying a forecast-time-safe PARC framework whose residual target, periodic features, same-feature controls, and auditable validation procedure improve short-horizon forecasting on the processed proxy series and remain competitive on public EV charging data while maintaining a clear information boundary. The framework is therefore best interpreted as a reproducible residual benchmarking approach for short-horizon operational forecasting, not as a universal performance claim for all EV-rich feeders.

Among the recurrent network benchmarks, LSTM outperformed Bi-LSTM on the main test interval. The strongest conclusion is not that a bidirectional recurrent network is inherently superior, but that a carefully specified residual-periodic forecasting problem can outperform more complex recurrent benchmarks on this short univariate series.

Funding: This research received no external funding.

Data Availability Statement: The original contributions presented in this study are included in the article. The Boulder EV transaction data used in this study were obtained from the City of Boulder Open Data portal, from the public “Electric Vehicle Charging Station Data” dataset. The dataset contains individual transaction records from city-owned EV charging stations in Boulder, Colorado, and was processed into a 15 min charging load series for the secondary EV validation case. Further inquiries can be directed to the corresponding author.

Conflicts of Interest: The author declares no conflicts of interest.

References

1. International Energy Agency. *Global EV Outlook 2025*; IEA: Paris, France, 2025. Available online: <https://www.iea.org/reports/global-ev-outlook-2025> (accessed on 16 May 2026).
2. Eren, Y.; Kucukdemiral, I. A comprehensive review on deep learning approaches for short-term load forecasting. *Renew. Sustain. Energy Rev.* **2024**, *189*, 114031. [CrossRef]
3. Sankalpa, C.; Kittipiyakul, S.; Laitrakun, S. Forecasting short-term electricity load using validated ensemble learning. *Energies* **2022**, *15*, 8567. [CrossRef]
4. Almaghrebi, A.; Aljuheshi, F.; Rafaie, M.; James, K.; Alahmad, M. Data-driven charging demand prediction at public charging stations using supervised machine learning regression methods. *Energies* **2020**, *13*, 4231. [CrossRef]
5. Zhu, J.; Yang, Z.; Guo, Y.; Zhang, J.; Yang, H. Short-term load forecasting for electric vehicle charging stations based on deep learning approaches. *Appl. Sci.* **2019**, *9*, 1723. [CrossRef]
6. Mohammad, F.; Kang, D.K.; Ahmed, M.A.; Kim, Y.C. Energy demand load forecasting for electric vehicle charging stations network based on ConvLSTM and BiConvLSTM architectures. *IEEE Access* **2023**, *11*, 67350–67369. [CrossRef]
7. Bamos, Z.N.; Laitos, V.M.; Afentoulis, K.D.; Vagropoulos, S.I.; Biskas, P.N. Electric vehicles load forecasting for day-ahead market participation using machine and deep learning methods. *Appl. Energy* **2024**, *360*, 122801. [CrossRef]
8. Narayana Gowda, S.; Nath, K.; Zhang, C.; Gowda, R.; Gadh, R. Short-term aggregate electric vehicle charging load forecasting in diverse conditions with minimal data using transfer and meta-learning. *Energy Syst.* **2026**, *17*, 347–366. [CrossRef]
9. Mansour, H.S.E.; Mohamed, A.S.; Abdel-Aziz, M. Electric vehicles charging stations load forecasting based on hybrid XGBoost-BiLSTM model. *Sci. Rep.* **2026**, *16*, 374. [CrossRef] [PubMed]
10. Kong, W.; Dong, Z.Y.; Jia, Y.; Hill, D.J.; Xu, Y.; Zhang, Y. Short-term residential load forecasting based on LSTM recurrent neural network. *IEEE Trans. Smart Grid* **2019**, *10*, 841–851. [CrossRef]

11. Chen, K.; Chen, K.; Wang, Q.; He, Z.; Hu, J.; He, J. Short-term load forecasting with deep residual networks. *IEEE Trans. Smart Grid* **2019**, *10*, 3943–3952. [[CrossRef](#)]
12. Hyndman, R.J.; Athanasopoulos, G. *Forecasting: Principles and Practice*, 3rd ed.; OTexts: Melbourne, Australia, 2021. Available online: <https://otexts.com/fpp3/> (accessed on 17 June 2026).
13. Cerqueira, V.; Torgo, L.; Mozetic, I. Evaluating time series forecasting models: An empirical study on performance estimation methods. *Mach. Learn.* **2020**, *109*, 1997–2028. [[CrossRef](#)]
14. Zhao, Q.; Liu, X.; Fang, J. Extreme gradient boosting model for day-ahead STLF in national level power system: Estonia case study. *Energies* **2023**, *16*, 7962. [[CrossRef](#)]
15. Yao, X.; Fu, X.; Zong, C. Short-term load forecasting method based on feature preference strategy and LightGBM-XGBoost. *IEEE Access* **2022**, *10*, 75257–75268. [[CrossRef](#)]
16. Li, Z.; Chen, Z. Short-term load forecasting based on CEEMDAN-FE-ISSA-LightGBM model. *Front. Energy Res.* **2023**, *11*, 1111786. [[CrossRef](#)]
17. Wang, Y.; Zhang, Z.; Pang, N.; Sun, Z.; Xu, L. CEEMDAN-CatBoost-SATCN-based short-term load forecasting model considering time series decomposition and feature selection. *Front. Energy Res.* **2023**, *10*, 1097048. [[CrossRef](#)]
18. Lee, J.; Cho, Y. National-scale electricity peak load forecasting: Traditional, machine learning, or hybrid model? *Energy* **2022**, *239*, 122366. [[CrossRef](#)]
19. Daniel, J.; Ayinde, K.; Lukman, A.F.; Albalawi, O.; Allohibi, J.; Alharbi, A.A. Optimised block bootstrap: An efficient variant of circular block bootstrap method with application to South African economic time series data. *AIMS Math.* **2024**, *9*, 30781–30815. [[CrossRef](#)]
20. Zhou, J.; Li, H.; Zhong, W. A modified Diebold-Mariano test for equal forecast accuracy with clustered dependence. *Econ. Lett.* **2021**, *207*, 110029. [[CrossRef](#)]
21. Wilcoxon, F. Individual comparisons by ranking methods. *Biom. Bull.* **1945**, *1*, 80–83. [[CrossRef](#)]

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.