

Supplementary Material

Supplementary Table S1. Hyperparameters and training settings of the prediction models

Note: The table reports the principal preprocessing, validation, hyperparameter, and baseline settings used in the experiments. Parameters that were not explicitly specified in the implementation were retained at package defaults.

| Model or setting         | Parameter            | Value                                 | Rationale                                                                                                                                     |
|--------------------------|----------------------|---------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------|
| Common target processing | Target variable      | Differential energy consumption (kWh) | The prediction target was constructed from first-order differences of cumulative electricity readings to represent short-period load changes. |
| Common target processing | Negative differences | Set to zero                           | Negative differences were treated as meter-reading regressions or communication artefacts rather than real negative energy consumption.       |
| Common target processing | Spike handling       | P99.5 percentile truncation           | Rare extreme spikes were truncated to reduce their influence on model training while retaining the sparse structure of the target series.     |
| Common validation        | Evaluation metrics   | MAE, RMSE, R <sup>2</sup> , SMAPE     | The same metrics were used to compare absolute error, sensitivity to large errors, explained variance, and relative error.                    |
| Ridge regression         | Feature scaling      | StandardScaler                        | Used to remove scale differences among variables and prevent high-magnitude features from dominating coefficient estimation.                  |
| Ridge regression         | $\alpha$             | 1.0                                   | Selected after comparing validation errors within the candidate range to balance coefficient shrinkage and prediction stability.              |
| Random Forest            | n_estimators         | 200                                   | The ensemble variance stabilizes as the number of trees increases; 200 trees provide a balance between stability and training cost.           |
| Random Forest            | random_state         | 42                                    | A fixed random seed was used to improve reproducibility.                                                                                      |
| Random Forest            | max_depth            | None                                  | Tree depth was not manually limited so that the model could retain nonlinear representation capacity.                                         |
| Random Forest            | min_samples_split    | 2                                     | The default minimum number of samples required for a split was retained to avoid overly restrictive tree growth.                              |
| Random Forest            | min_samples_leaf     | 1                                     | The default minimum number of samples at a leaf node was retained to avoid premature restriction of local fitting.                            |
| Random Forest            | criterion            | squared_error                         | Mean squared error was used as the splitting criterion for the continuous regression target.                                                  |
| Random Forest            | bootstrap            | True                                  | Bootstrap sampling was enabled to increase diversity among trees and reduce correlation within the ensemble.                                  |
| Random Forest            | max_features         | 1.0 (all features)                    | All features were allowed to participate in candidate splits, consistent with the default scikit-learn regression setting.                    |
| Baseline XGBoost         | Cross-validation     | KFold(n_splits = 5, shuffle = False)  | Used as a baseline boosted-tree setting that preserves sample order without a rolling time-series validation mechanism.                       |
| Baseline XGBoost         | n_estimators         | 500                                   | A larger number of boosting rounds was combined with a relatively small learning rate to iteratively correct residuals.                       |
| Baseline XGBoost         | learning_rate        | 0.05                                  | A small step size was used to avoid overly large single-round updates on the noisy differential target.                                       |
| Baseline XGBoost         | max_depth            | 6                                     | This setting balances nonlinear interaction modeling and tree-complexity control.                                                             |
| Baseline XGBoost         | subsample            | 0.8                                   | Sample subsampling was used to reduce inter-tree correlation and mitigate overfitting.                                                        |
| Baseline XGBoost         | colsample_bytree     | 0.8                                   | Feature subsampling was used to reduce the amplification of redundant or highly correlated features.                                          |
| Baseline XGBoost         | reg_alpha            | Not explicitly specified (default)    | The L1 regularization term was retained at the package default because it was not separately specified in the implementation.                 |
| Baseline XGBoost         | reg_lambda           | Not explicitly specified (default)    | The L2 regularization term was retained at the package default because it was not separately specified in the implementation.                 |

| Model or setting                  | Parameter                             | Value                                       | Rationale                                                                                                             |
|-----------------------------------|---------------------------------------|---------------------------------------------|-----------------------------------------------------------------------------------------------------------------------|
| Baseline XGBoost                  | random_state                          | 42                                          | A fixed random seed was used to improve reproducibility.                                                              |
| Optimized XGBoost workflow        | Cross-validation                      | TimeSeriesSplit(n_splits = 5)               | Temporal order was preserved to reduce future-information leakage and better match the forecasting setting.           |
| Optimized XGBoost workflow        | eta                                   | 0.04                                        | A smaller learning rate was used to improve robustness on the sparse differential target.                             |
| Optimized XGBoost workflow        | max_depth                             | 5                                           | Tree depth was slightly reduced to limit the influence of rare spike samples on tree structure.                       |
| Optimized XGBoost workflow        | min_child_weight                      | 12                                          | A stronger constraint on child-node splitting was used to suppress overfitting.                                       |
| Optimized XGBoost workflow        | subsample                             | 0.8                                         | Subsampling was used to improve stability across temporal folds and reduce the effect of sample noise.                |
| Optimized XGBoost workflow        | colsample_bytree                      | 0.8                                         | Feature subsampling was used to reduce the dominance of highly correlated features in individual trees.               |
| Optimized XGBoost workflow        | reg_lambda                            | 3                                           | Stronger L2 regularization was used to reduce overfitting to rare spike samples.                                      |
| Optimized XGBoost workflow        | objective                             | reg:squarederror                            | Squared-error regression was used for continuous-value prediction.                                                    |
| Optimized XGBoost workflow        | eval_metric                           | rmse                                        | RMSE was used as the validation metric because it is sensitive to larger errors.                                      |
| Optimized XGBoost workflow        | tree_method                           | hist                                        | The histogram-based method was used to improve training efficiency on large samples.                                  |
| Optimized XGBoost workflow        | early_stopping_rounds                 | 120                                         | Training was stopped when validation error no longer improved for consecutive rounds, reducing overfitting risk.      |
| Optimized XGBoost workflow        | num_boost_round                       | 1500                                        | A large upper bound was set while relying on early stopping to determine the effective number of boosting rounds.     |
| Optimized XGBoost workflow        | Validation ratio within training fold | 0.15                                        | The last portion of each training fold was used for early stopping and validation monitoring.                         |
| Optimized XGBoost workflow        | Time-derived features                 | Enabled                                     | Hour, weekday, and weekend indicators were included to represent periodic operating rhythms.                          |
| Optimized XGBoost workflow        | Lag features                          | [1, 2, 5, 10, 30]                           | Multiple lag orders were used to represent short-term dependence in the differential energy series.                   |
| Optimized XGBoost workflow        | Rolling windows                       | [3, 5, 15, 30]                              | Rolling means and standard deviations were used to represent local smoothing and background load at different scales. |
| Optimized XGBoost workflow        | Nonzero windows                       | [5, 30, 60]                                 | Recent nonzero-count features were used to identify active energy-change periods in the sparse target series.         |
| Optimized XGBoost workflow        | seed                                  | 42                                          | A fixed seed was used to improve reproducibility.                                                                     |
| Recurrent neural network baseline | Input scaling                         | StandardScaler                              | Standardization was used to maintain effective gradient scales under the sparse differential target.                  |
| Recurrent neural network baseline | Target scaling                        | StandardScaler                              | The output target was standardized to improve training stability.                                                     |
| Recurrent neural network baseline | Time steps                            | 24                                          | A 24-step sliding window was used to capture intraday dependence suggested by the time-series analysis.               |
| Recurrent neural network baseline | n_splits                              | 5                                           | The same number of temporal validation folds was used for consistency with the other models.                          |
| Recurrent neural network baseline | Hidden units                          | 64/32                                       | A moderate-capacity stacked recurrent structure was used to balance representation capacity and training stability.   |
| Recurrent neural network baseline | Dropout                               | 0.2                                         | Moderate dropout was used to suppress overfitting without excessively weakening temporal representation.              |
| Recurrent neural network baseline | Optimizer                             | Adam                                        | Adam was used for gradient-based optimization because of its stable convergence on nonstationary targets.             |
| Recurrent neural network baseline | Learning rate                         | $1.0 \times 10^{-3}$                        | This value was used as a stable initial learning rate and combined with learning-rate decay.                          |
| Recurrent neural network baseline | Batch size                            | 128                                         | The batch size was selected to balance training efficiency and stability in the implementation.                       |
| Recurrent neural network baseline | Epochs                                | 30                                          | The maximum number of epochs was limited because early stopping was used to avoid unnecessary training.               |
| Recurrent neural network baseline | Early stopping                        | Enabled                                     | Early stopping restored the best validation weights and reduced overfitting risk.                                     |
| ARIMA baseline                    | Input variables                       | Univariate differential energy series only  | ARIMA was used to test the autoregressive predictability of the target series without environmental variables.        |
| ARIMA baseline                    | d                                     | 0                                           | The differenced energy target was already used; therefore, additional differencing was not applied.                   |
| ARIMA baseline                    | Candidate orders                      | (1,0,0), (2,0,0), (1,0,1), (2,0,1), (2,0,2) | Low-order candidates were used to avoid excessive complexity on the sparse target series.                             |

| Model or setting         | Parameter       | Value                                   | Rationale                                                                                                            |
|--------------------------|-----------------|-----------------------------------------|----------------------------------------------------------------------------------------------------------------------|
| ARIMA baseline           | Order selection | AIC/BIC                                 | Information criteria were used to select the order within each temporal validation fold.                             |
| Historical mean baseline | Prediction rule | Training-fold mean of the target series | This baseline provides a global mean-prediction reference without temporal dynamics.                                 |
| Naive last baseline      | Prediction rule | Previous observed target value          | This baseline tests whether one-step persistence can represent short-term inertia in the differential energy series. |