

Article

A Robust Model Evaluation Process for Early-Stage Cooling Load Prediction of Buildings

Yaren Aydın ¹ , Ümit Işıkdag ² , Sinan Melih Nigdeli ¹ , Gebrail Bekdaş ^{1,*} , Wook-Won Kim ³ and Zong Woo Geem ^{3,*} 

¹ Department of Civil Engineering, Istanbul University-Cerrahpasa, Istanbul 34320, Türkiye; yaren.aydin@iuc.edu.tr (Y.A.); melihnig@iuc.edu.tr (S.M.N.)

² Department of Architecture, Mimar Sinan Fine Arts University, Istanbul 34427, Türkiye; umit.isikdag@msgsu.edu.tr

³ Department of Smart City, Gachon University, Seongnam 13120, Republic of Korea; wookwonkim@gachon.ac.kr

* Correspondence: bekdas@iuc.edu.tr (G.B.); geem@gachon.ac.kr (Z.W.G.)

Abstract

In the construction industry, a large portion of energy is spent on heating and cooling, which both increases costs and contributes to resource depletion. The aim of the study was to provide and evaluate a robust ML model evaluation process for early design stage cooling load prediction of buildings. For this purpose, 18 different machine learning models were evaluated using a Nested Cross-Validation approach consisting of 50 outer fold and 50 inner Optuna trials, along with hyperparameter optimization. To avoid model selection being dependent on small decimal differences, paired model comparisons, effect sizes, Holm-corrected statistical tests, and the 1-SE economy rule were applied over the same outer folds. As a result of the analysis, Categorical Boosting (CatBoost) was selected as the final model, and within the Nested-CV framework, $R^2 = 0.8275 \pm 0.0072$, $RMSE = 1.6792 \pm 0.0210$ kWh, $MAE = 1.4356 \pm 0.0233$ kWh, and $MAPE = 0.0521 \pm 0.0009$ were obtained. Model interpretability analyses showed that the variables Ambient Temperature, Solar Radiation, and Heat Reflective Treatment had the highest permutation importance values. Residual analyses revealed that the model exhibited low systematic bias, but the residual variance was dependent on the estimate value, and the residuals deviated from a normal distribution. This study provides a framework that evaluates not only the prediction performance but also model selection, generalization stability, interpretability, and residual behavior together. The findings demonstrate that CatBoost is a strong option for cooling load prediction in this simulation-based dataset. However, validation of the obtained results with real building data and different climatic conditions is considered an important requirement for future studies in terms of evaluating the external validity of the model.

Keywords: building energy efficiency; cooling load; Optuna; Nested CV; machine learning

Academic Editors: Maria Dolores Duran Garcia, Eduardo González-Mora and Artur O. Zaporozhets

Received: 22 July 2026

Revised: 13 August 2026

Accepted: 19 August 2026

Published: 20 August 2026

Copyright: © 2026 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the [Creative Commons Attribution \(CC BY\)](https://creativecommons.org/licenses/by/4.0/) license.

1. Introduction

Today, designs aimed at improving the efficiency of energy for buildings and ensuring sustainable thermal comfort conditions indoors are becoming increasingly important in the fields of architecture and engineering.

In 2024, total final energy consumption exceeded 450 EJ, an increase of approxi-

mately 25 EJ compared to 2019. Buildings accounted for approximately 30% of global energy demand. In developed economies, approximately 70% of residential energy is used for space and water heating, while in developing economies, per capita heating energy use is lower. However, due to population growth, rising incomes, and increasing temperatures, energy demand, especially from cooling, is rapidly increasing [1]. The average annual growth rate of energy demand was 1.3% during 2013–2023, increased to 2.0% in 2024, and was 1.3% in 2025 [2].

In 2025, China contributed the most to the increase in global energy demand. However, the growth rate of energy demand slowed to 1.7%, down from 2024. Energy demand growth in the United States accelerated, exceeding 2% in 2025. The United States accounted for approximately one-fourth of the increase in global energy demand [2]. The rapid depletion of energy resources on a global scale highlights the importance of developing sustainable energy systems and increasing energy efficiency.

The building sector generates high energy demands both during the construction process and the usage phase. Heating, cooling, ventilation, and lighting systems constitute a large portion of the total energy consumption in buildings [3]. Heat losses through the building envelope are one of the main causes of energy consumption. Therefore, one of the most effective methods for increasing energy efficiency is thermal insulation. Thermal insulation reduces energy consumption by limiting heat transfer through building elements [4]. Ensuring indoor comfort is of great importance in terms of user satisfaction and energy efficiency. Thermal comfort, in particular, directly affects the health and performance of users [5].

Facades play a crucial role in maintaining comfortable indoor conditions by forming a boundary between the variable weather conditions of the external environment and the interior space. In this respect, facade systems are an important building component that affects energy efficiency in buildings [6]. Population growth and technological advancements are driving up energy demand [7]. Rising energy demand and climate change have made the building sector a priority area for intervention in terms of energy efficiency [8].

Building energy performance is affected by the interaction of numerous parameters [9]. A building's energy performance is influenced by factors such as climatic conditions, the building envelope, energy systems, operation and maintenance practices, user behavior, and indoor environmental quality [10]. The building envelope, as a fundamental component regulating the building's relationship with its external environment, directly affects energy performance. These findings necessitate addressing loads such as heating, cooling, and lighting in order to reduce energy consumption in buildings. Because buildings account for a significant portion of energy consumption, energy efficiency has become a top priority in design.

Data-driven techniques are widely used to estimate heating and cooling loads in buildings in order to maximize energy efficiency [11]. A building's surface area and the dimensions of its walls and windows are key factors affecting energy consumption. The heating load represents the amount of energy required to maintain the indoor temperature, while the cooling load represents the amount of energy that must be removed from the building [12].

Data-driven approaches, based on extracting meaningful patterns from measurement data, offer a significant alternative in predicting the behavior of complex physical systems [13,14].

Artificial intelligence (AI) is an interdisciplinary field of research aimed at enabling computer systems to perform human-like cognitive processes such as learning, reasoning, decision-making and problem-solving. With the increase in the volume of data and

computational capacity in recent years, AI methods such as machine learning and deep learning have come to the fore and have begun to be widely used in many sectors, including energy [15], education [16] and healthcare [17]. Machine learning is a subfield of AI that enables computers to make predictions and classifications by learning from data, which is the main technical method implemented in this research.

Unlike most previous studies, this study minimized optimization bias by performing hyperparameter optimization within a repeated nested cross-validation process and provides a more reliable and statistically robust evaluation process for cooling load estimation by reporting model performance using bootstrap-based confidence intervals.

Model performance was evaluated not only using mean error metrics but also using bootstrap-based 95% confidence intervals, thereby providing a descriptive characterization of the variability of the observed outer-fold performance scores.

By performing hyperparameter optimization in each outer layer exclusively on the relevant training data, the study prevented data leakage, and an unbiased evaluation of generalization performance was ensured. A comprehensive comparison was conducted, covering various linear, nonlinear, ensemble, and gradient-boosting-based algorithms. This study not only delivers high prediction accuracy but also proposes a robust, unbiased, and reproducible evaluation process that can be utilized in future building energy prediction studies.

1.1. Literature Review

Machine learning is widely used in predictive modelling of energy systems and has become an important area of research in recent years, particularly in efforts to improve building energy efficiency. In particular, data-driven approaches are increasingly being preferred to determine the effects of building characteristics on the cooling load and to accurately estimate the cooling load. In this context, a literature review was conducted to identify current research related to the context of this study. In the following, current studies representing different approaches to the subject are presented in chronological order. Although numerous studies exist in the literature regarding building cooling load estimation, this section primarily addresses data-driven methods and machine learning-based approaches found in the literature.

Early studies include that by Fan and Ding (2019) [18] who developed a multiple nonlinear regression (MNR) model to estimate the short-term cooling load in public buildings. The case study conducted showed that the calibrated MNR model provided higher accuracy ($R^2 = 0.958$, RMSE = 405.7 kW) than initial estimates and traditional regression models. Sajjad et al. (2020) [12] proposed a new gated recurrent unit (GRU)-based framework that uses multi-output learning to simultaneously predict heating and cooling loads and achieved superior performance (MAE = 0.0167, MSE = 0.0006, RMSE = 0.0247, MAPE = 0.0368) compared to existing machine learning and deep learning models on an energy efficiency dataset. Li et al. (2021) [19] proposed a new Recurrent Neural Network (RNN) architecture with an attention mechanism to predict building cooling loads 24 h in advance. They visualized the model's decision-making process using attention vectors, and the model achieved higher accuracy (RMSE = 524, MAE = 377) and interpretability compared to basic RNN models. Chaganti et al. (2022) [20] proposed a machine learning model that provides high accuracy for estimating cooling and heating loads and examined the effect of building characteristics on the model's predictive performance. The results showed that relative compactness, surface area, and wall area were the most influential variables, and the proposed model achieved an R^2 of 0.999 for heating load and an R^2 of 0.997 for cooling load. Kang et al. (2022) [21] proposed an integrated cooling load prediction and optimized control approach for ice-based thermal en-

ergy storage (TES) systems in commercial buildings. They supported the prediction model they developed with a midday update mechanism and integrated it with a control strategy tailored to time-of-use pricing. In an application conducted in an actual commercial building, the model achieved a mean absolute error (MAE) of 389 kW and a coefficient of variation (CV) of 12.5%. Lu et al. (2023) [22] proposed an AutoML-based framework to quickly and accurately predict a building's energy performance in the early stages of design. In evaluations conducted on an experimental residential dataset, the proposed method achieved high accuracy, with $R^2 = 0.9965$ and $RMSE = 0.602$ kWh/m² for heating load, and $R^2 = 0.9899$ and $RMSE = 0.973$ kWh/m² for cooling load. Alyakoob et al. (2023) [23] evaluated various machine learning algorithms using micro-climate data generated with ENVI-met to estimate the cooling loads of three university buildings in Arizona, and achieved the most successful result with Random Forest (RF), with 98% accuracy. Guo et al. (2023) [24] developed four hybrid models by combining the LightGBM model with the Random Search, Grid Search, CMA-ES, and TPE hyperparameter optimization methods. The results showed that the TPE-LightGBM model provided the highest accuracy in both heating and cooling load forecasting ($R^2 = 0.9981$ for heating; $R^2 = 0.9924$ for cooling). Mehdizadeh Khorrami et al. (2024) [25] predicted the heating and cooling loads for 12 different building types based on their area and height characteristics using linear regression, decision trees, logistic regression, and neural networks. Of these models, the decision tree model proved to be the most successful, achieving accuracies of 98.96% and 93.24% for heating and cooling loads, respectively. Gao et al. (2024) [26] used SVR, XGBoost, and a Dempster–Shafer-based ensemble model on a dataset consisting of 768 samples and optimized the models using the Coot, Dingo, and Sea-Horse optimization algorithms. The results show that the XGBoost-Sea-Horse model, optimized with the Sea-Horse Optimizer, achieved the highest accuracy ($R^2 = 99.36\%$) and the lowest RMSE (0.7723) for both heating and cooling load predictions. Aydın et al. (2025) [27] compared 16 different machine learning algorithms to predict the total heating and cooling loads of buildings with ultra-high-performance fiber-reinforced concrete (UHP-FRC) facade panels. The models achieved high accuracy, with R^2 values above 0.99. Furthermore, low error values (MAE = 25.46; RMSE = 53.87; NMAE = 0.0012; NRMSE = 0.0027) demonstrated the reliability of the predictions. Peng (2025) [28] developed a RF model to estimate cooling loads in residential buildings and improved its performance using the Tasmanian Devil Optimization (TDO) and Northern Goshawk Optimization (NGO) algorithms. The RF-TDO model achieved the best results, with $R^2 = 0.997$ and $RMSE = 0.498$. Zhang et al. (2026) [29] proposed a hybrid Gray Wolf Optimization (GWO)-CNN-LSTM model for cooling load estimation. As a result, the proposed GWO-CNN-LSTM model achieved values of 0.9266 R^2 , 218.78 W MAE, and 327.40 W RMSE.

1.2. Research Significance

This study developed a computationally efficient surrogate machine learning model to approximate the outputs of a physics-based digital twin thermal simulation for building cooling load prediction. Rather than discovering new physical relationships, the proposed model serves as a surrogate for computationally intensive simulations, enabling rapid cooling-load estimation during iterative building design, optimization, and decision-support processes.

The primary contribution of this study is not only the development of an accurate predictive model but also the establishment of a statistically rigorous evaluation process. Unlike many previous studies that relied on a single training–test split or conventional cross-validation, this work employed repeated five-fold nested cross-validation with ten

repetitions. In our process, hyperparameter optimization and performance evaluation were separated through the inner and outer loops; therefore, optimistic bias associated with hyperparameter tuning was substantially reduced. Hyperparameter optimization was performed at the inner loop of each outer-training partition using Optuna's Tree-structured Parzen Estimator (TPE), where 50 optimization trials were evaluated by inner five-fold cross-validation. The hyperparameter search spaces for the 18 models evaluated in this study were determined using Optuna's Tree-structured Parzen Estimator (TPE) sampling method with 50 trials and five-fold cross-validation in inner loops, and the details of the search spaces for each model are presented in Supplementary Materials Table S2.

The final performance evaluation was based exclusively on the 50 outer-test evaluations. Consequently, the reported results provide a nested-cross-validation estimate of out-of-sample performance for observations drawn from the same simulation-generated distribution. Inner-fold results were intentionally excluded from final performance reporting because they participated in hyperparameter optimization and therefore tend to produce optimistic estimates.

To provide reproducible and leakage-controlled preprocessing, all data transformations—including ordinal encoding, one-hot encoding, and standardization where required—were implemented within the scikit-learn Pipeline and fitted separately within each training fold. This prevented the corresponding outer-test fold from being used when fitting preprocessing transformations or the model. Only predictors available during the intended early design stage were retained, while variables directly reflecting cooling load or generated after simulation were excluded.

In this study, eighteen regression algorithms were evaluated under the identical experimental protocol. Model performance was assessed using four complementary metrics: R^2 , RMSE, MAE, and MAPE. Rather than relying solely on R^2 , the final model was selected according to the lowest out-of-sample RMSE because it directly reflects prediction error magnitude and penalizes large errors more heavily. Performance metrics are reported as mean $\pm$ standard deviation across the 50 outer test folds, and 95% percentile-bootstrap confidence intervals for the mean performance metric were calculated using 10,000 resamples of the 50 outer-fold metric values. This provides an estimate of the variability of model performance across repeated data samplings rather than relying on a single performance value.

Model interpretability was intentionally separated from performance evaluation. After model selection, the best-performing model was retrained on the complete dataset solely for permutation importance analysis to determine feature importance. The performance obtained from this retrained model was not reported or used to assess predictive accuracy, ensuring that interpretability analyses did not influence estimates of generalization performance.

Among the evaluated algorithms, CatBoost achieved the lowest mean nested CV RMSE, with an average R^2 of 0.8275 ± 0.0072 , RMSE of 1.6792 ± 0.0210 kWh, MAE of 1.4356 ± 0.0233 kWh, and MAPE of 0.0521 ± 0.0009 .

These results provide a nested-cross-validation estimate of the model's generalization performance within the simulation-generated data distribution.

The dataset used in this study is the "Heat Reflective Building Energy Dataset" provided by Colabsss on the Kaggle [28]. According to the dataset provider's description, the dataset was created for digital twin-based thermal performance analysis and includes environmental conditions, building thermal properties, HVAC settings, and energy performance indicators. However, the detailed physical structure of the Digital Twin model used in the existing dataset documentation, the validation process using ac-

tual measurements, and the physical constraints applied in the creation of input variables are not explained. Therefore, the physical consistency of the input combinations in the dataset or the experimental validation of the underlying Digital Twin model was not assumed within the scope of this study. The dataset was treated as a data source representing the existing simulation-based input–output relationship defined by the provider, and machine learning models were evaluated as surrogate models for cooling load prediction within this data domain.

Overall, this study contributes a comprehensive surrogate modelling process that focuses on the early design stage, combining rigorous nested cross-validation, strict separation of hyperparameter optimization and performance evaluation, multiple complementary performance metrics, statistical uncertainty quantification through bootstrap confidence intervals, and model interpretability. Together, these methodological choices support a more reproducible and carefully controlled evaluation of machine learning models for building energy prediction.

2. Materials and Methods

In this study, nested cross-validation with 5×10 iterations was used to evaluate the performance of 18 ML models. The flowchart of the study is given in Figure 1. Hyperparameter optimization was performed only on the training data of each outer layer, while performance evaluation was based on the results obtained from a total of 50 outer test folds. All data preprocessing steps were executed within a scikit-learn pipeline, and performance was determined using the R^2 , RMSE, MAE, and MAPE metrics, along with mean $\pm$ standard deviation and 95% bootstrap confidence intervals.

To objectively compare the performance of different algorithms, a total of 18 different models (RF, eXtreme Gradient Boosting (XGBoost), Light Gradient Boosting Machine (LightGBM), Categorical Boosting (CatBoost), Gradient Boosting, Histogram GradientBoosting, AdaBoost, Extra Trees, Decision Tree, Bagging, Linear Regression, Ridge, Lasso, Elastic Net, Huber, K-Nearest Neighbours (KNN), Support Vector Regression (SVR) and Multi-Layer Perceptron (MLP)) were evaluated during the research.

Figure 2 presents the BPMN (Business Process Model and Notation) diagram of the study. The process begins when the research objective is defined. The dataset, preprocessing rules and 18 candidate models are then registered as process inputs. This establishes a common evaluation protocol before model execution starts.

From a process perspective, the central activity is a multi-instance subprocess: “evaluate each candidate model.” It is executed once for every model. Within each execution, the outer cross-validation loop produces the generalization evidence, while the inner subprocess performs preprocessing, hyperparameter optimization and model selection using only the corresponding outer-training partition.

After inner tuning, the optimized model is refitted on the complete outer-training partition and evaluated once on the untouched outer-test partition. The exclusive gateway checks whether more outer splits remain. If so, execution returns to the next split. This continues until all 50 outer evaluations are completed.

The results are aggregated and passed to model comparison. A second gateway checks whether all 18 candidate models have completed the same process. The lowest mean RMSE determines the winner.

The final full-dataset tuning and refitting activity are intentionally positioned after model selection. It produces the deployable or interpretable final model, but it does not generate performance evidence. Generalization performance comes exclusively from the nested cross-validation path.

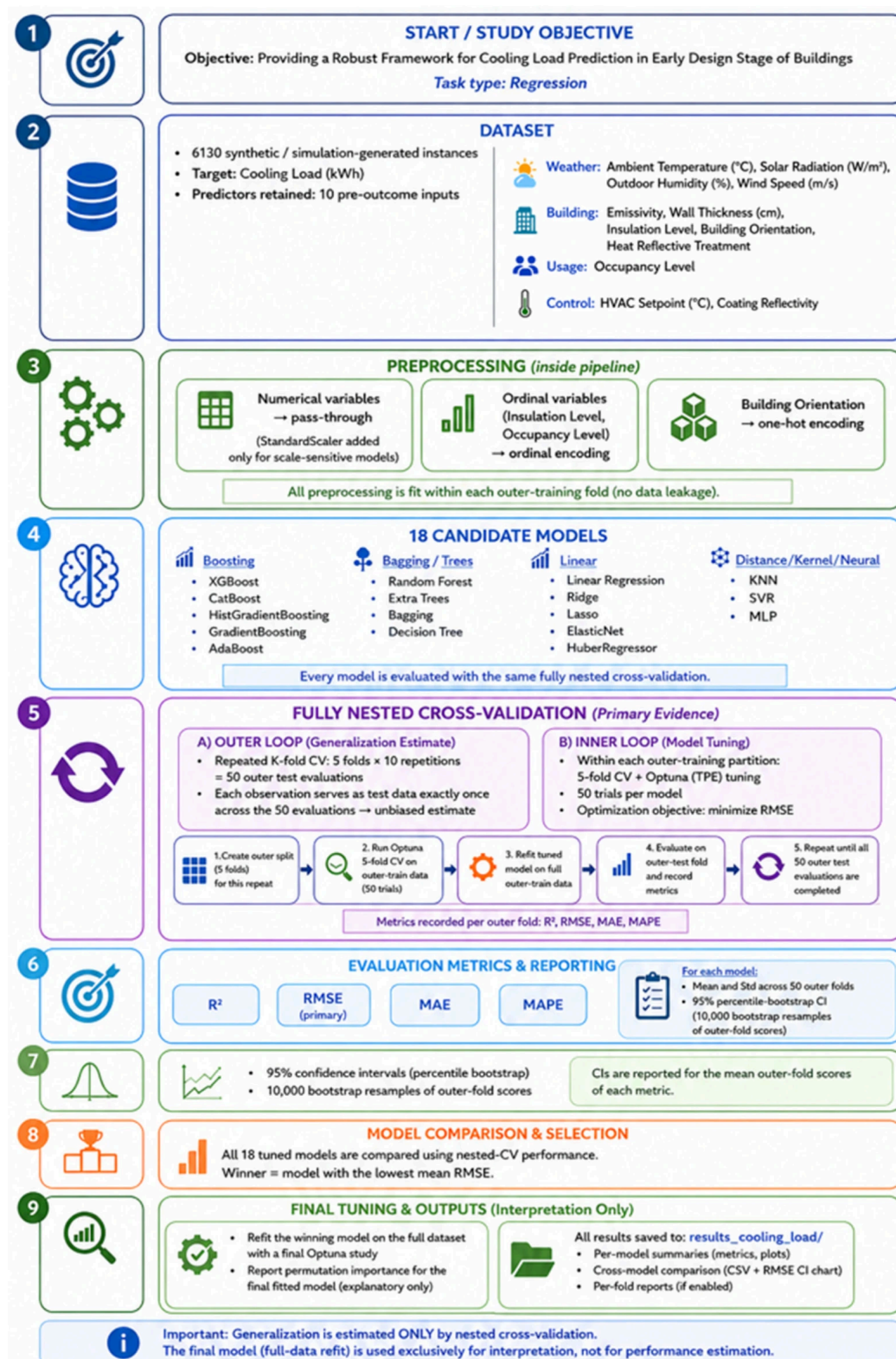


Figure 1. Machine learning pipeline for cooling load prediction. (Note: each observation serves once as outer-test data within each five-fold repetition and therefore ten times across the ten repetitions.).

The process lens makes the main control mechanisms visible, for example by ensuring preprocessing is contained within training partitions, outer-test data remain isolated until evaluation, every candidate follows the same process, and model selection occurs only after all repeated evaluations have finished.

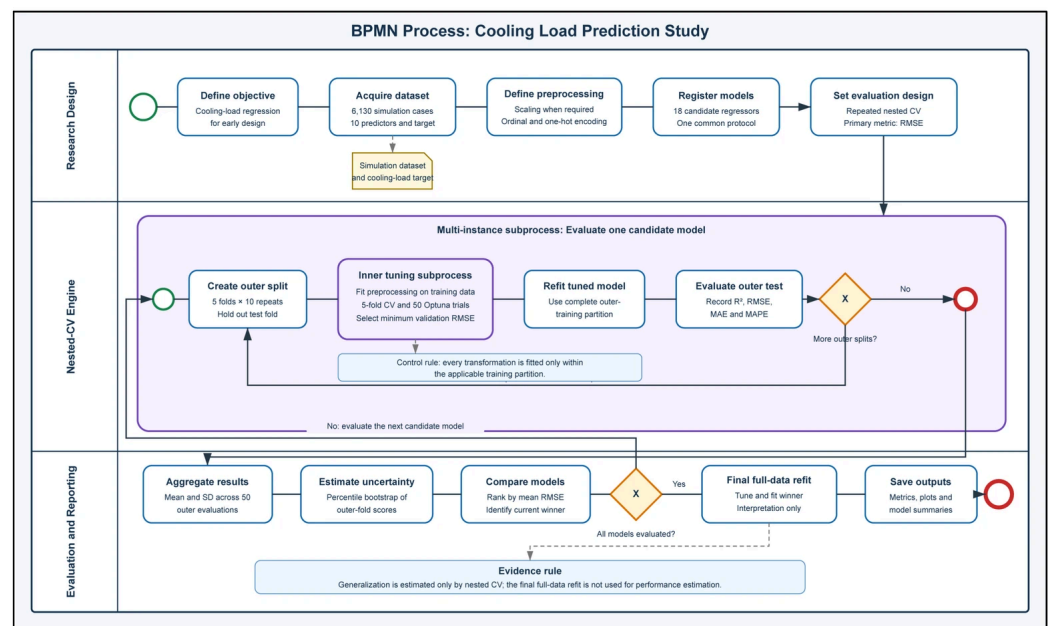


Figure 2. The process lens: BPMN diagram of the study.

Figures 1 and 2 serve complementary rather than interchangeable purposes. Figure 1 provides a methodological overview of what components constitute the machine-learning evaluation framework, whereas Figure 2 formalizes how these components are executed and controlled as a process. The BPMN representation makes explicit the execution logic that is only summarized in the flowchart, including repeated subprocess execution for each candidate model, decision gateways controlling progression through outer folds and model comparison, separation of tuning and testing activities, completion conditions, and the positioning of full-dataset refitting after performance-based model selection. Thus, the BPMN model does not introduce an additional analytical method; rather, it represents the experimental protocol as a repeatable and auditable process with explicit control points.

2.1. Dataset Source and Dataset Description

The Heat Reflective Building Energy Dataset [30] was obtained from the Kaggle database. The dataset consisted of a total of 6130 rows. The dataset was created to examine the impact of heat-reflective surface treatments applied to buildings on energy efficiency. The dataset examines the impact of heat-reflective surface treatments on building facades on energy efficiency through digital twin-based thermal performance analysis [30]. The environmental conditions, building thermal properties, HVAC parameters, and energy performance indicators included in this dataset also provide a suitable process for the prediction of the cooling load (kWh).

Since the dataset is maintained by its original authors and may be modified or removed from Kaggle, a snapshot of the exact version used in this study has been archived in Zenodo for reproducibility and long-term accessibility. This archived copy does not claim ownership of the original dataset and is provided solely as a reference to the version used in this study [31].

In the dataset, variables representing thermal-response outcomes, energy outcomes, simulation-derived indicators, or treatment information that were unavailable at the intended prediction stage were excluded. Energy Consumption (kWh), Peak Load (kW), Surface Temperature Reduction (°C), Indoor Comfort Index, Wall Surface Temperature (°C), Roof Surface Temperature (°C), Indoor Temperature (°C), and Coating Reflectivity

(%) were excluded because they represent post-outcome or simulation-derived information. Heat Reflective Treatment was additionally excluded because it represents a simulation scenario or treatment label that may directly encode the data-generating condition.

The original predictor set excluded and Coating Reflectivity (%) and Heat Reflective Treatment under the same rule used to remove genuine outcome variables (e.g., indoor temperature, energy consumption). On review, this was not appropriate: unlike those variables, which are physical measurements that only exist after the cooling process occurs, coating reflectivity and treatment status are building design specifications fixed *before* construction, independent of any cooling event—the same category as insulation level and wall thickness, which were already included. To confirm whether these two variables carry real, independent information rather than just restating something the model already knows, we tested whether they could be predicted from the other ten predictors (they could not, $R^2 \approx 0$) and whether they added explanatory power once everything else was controlled for (they did: cross-validated R^2 rose from 0.771 to 0.820, a gain that held up and did not disappear under scrutiny, unlike a related variable, surface temperature reduction, which looked informative at first but turned out to just be a downstream footprint of the treatment itself and was correctly left out). On this basis, Coating Reflectivity (%) and Heat Reflective Treatment were reclassified as legitimate predictors and added to the model.

The input variables of the dataset were Ambient Temperature (°C), Solar Radiation (W/m^2), Outdoor Humidity (%), Wind Speed (m/s), Emissivity, Wall Thickness (cm), Insulation Level, Building Orientation, Occupancy Level, HVAC Setpoint (°C), Coating Reflectivity (%) and Heat Reflective Treatment. The output variable was the Cooling Load (kWh).

In the dataset, each row represented a scenario for a planned building. The inputs and output used in this study are shown in Table 1.

Table 1. Input variables.

Group	Feature	Type
<i>Input</i>		
Weather	Ambient Temperature (°C)	Numeric
	Solar Radiation (W/m^2)	Numeric
	Outdoor Humidity (%)	Numeric
	Wind Speed (m/s)	Numeric
Building	Emissivity	Numeric
	Wall Thickness (cm)	Numeric
	Insulation Level	Ordered category (Low/Medium/High)
	Building Orientation	Category (N/E/S/W)
	Coating Reflectivity (%)	Numeric
Usage	Occupancy Level	Ordered category (Low/Medium/High)
Control	HVAC Setpoint (°C)	Numeric
	Heat Reflective Treatment	Category (Applied/Not Applied)
<i>Output</i>		
	Cooling Load (kWh)	Numeric

Table 2 shows descriptive statistics for the variables with numerical features.

Table 2. Descriptive statistics for variables with numerical features.

Feature	Count	Mean	Std	Min	Median	Max	Skewness	Kurtosis
<i>Inputs</i>								
Ambient Temperature (°C)	6130	36.449	4.921	28	36.465	45	0.011	−1.205
Solar Radiation (W/m ²)	6130	772.031	186.529	450.03	769.555	1099.79	0.023	−1.183
Outdoor Humidity (%)	6130	65.101	14.431	40.01	65.23	90	−0.014	−1.195
Wind Speed (m/s)	6130	4.333	2.167	0.5	4.36	8	−0.039	−1.189
Emissivity	6130	0.851	0.057	0.75	0.85	0.95	−0.03	−1.161
Wall Thickness (cm)	6130	25.018	5.745	15	24.98	35	−0.003	−1.188
HVAC Setpoint (°C)	6130	23.016	1.723	20	23.06	26	−0.005	−1.208
Coating Reflectivity (%)	6130	77.303	10.18	60	77.135	95	0.033	−1.234
<i>Output</i>								
Cooling Load (kWh)	6130	28.4	4.047	16.8	28.41	39.88	−0.015	−0.464

An examination of Table 2 reveals that the features in the dataset exhibit different scales and distribution structures. The distribution was more symmetrical for variables where the mean and median values are close to each other. Standard deviation values were quite low for some variables. Most skewness values are very close to zero, and it appears that the numerical variables generally exhibit a distribution close to a symmetric one. This indicates that a significant portion of the variables do not exhibit right or left skewness and that the distribution is balanced. Kurtosis values, on the other hand, were negative for all variables and ranged from approximately −0.464 to −1.208. In other words, the concentration of outliers in the variables is low. Overall, the numerical columns in the dataset appear to be largely symmetrically distributed. This indicates that outliers were limited in these variables and that the variables exhibited a more balanced distribution.

Table 3 shows the values of the categorical predictors. Insulation Level and Occupancy Level were ordinal encoded, whereas Building Orientation was one-hot encoded.

Table 3. Values for categorical variables.

Feature	Values
<i>Inputs</i>	
Insulation Level	Low, Medium, High
Building Orientation	South, West, East, North
Occupancy Level	Low, Medium, High
Heat Reflective Treatment	Applied, Not Applied

Data Preprocessing

To provide a consistent numerical representation across all candidate algorithms, categorical predictors were transformed using ordinal or one-hot encoding within the common preprocessing pipeline [32]. Insulation Level and Occupancy Level were encoded as ordered categories, whereas Building Orientation was represented by one-hot indicators.

Ordinal categories in the dataset, such as Insulation Level and Occupancy Level, have been converted to numerical values using ordinal encoding (Low = 0, Medium = 1, High = 2). The Building Orientation variable, on the other hand, has been converted into

separate columns using the one-hot encoding method.

For models requiring standardization (linear models, K-Nearest Neighbours (KNN), Support Vector Regression (SVR), Neural Networks), the variables have been scaled. In this way, each feature has been rescaled to a comparable range. All preprocessing was done only on training folds.

Ordinal encoding assigns ordered integer values to categories with an inherent ranking. In this study, Insulation Level and Occupancy Level were encoded as Low = 0, Medium = 1, and High = 2. Building Orientation contained four nominal categories and was one-hot encoded with one reference category omitted, resulting in three binary indicator columns. One-hot encoding ordinarily creates one indicator for each category; using a reference category yields $d - 1$ indicators [33,34].

One-hot coding transforms a single variable with n observations and d distinct values into d binary variables, each with n observations. Each observation indicates the presence (1) or absence (0) of the binary variable.

2.2. The Preprocessing Pipeline

Owing to the developed pipeline, data preparation steps (encoding and, if necessary, standardization) and the machine learning model were integrated into a single process. Since the pipeline integrates data preparation processes with the machine learning model within a single system, the same operations are automatically applied in the same order at each cross-validation fold. All transformations were applied exclusively on the training data at each cross-validation fold and then applied to the test data to reliably evaluate model performance.

The pipeline consists of three preparatory steps. (1) The numeric variables pass through (2) ordinal encoding and (3) one-hot encoding. For models sensitive to the scale of the numbers (linear models, nearest neighbors, support vector machines, neural networks), an additional step (standardization) has been added to rescale each feature to a comparable range. This step was not necessary for tree-based models.

2.3. Evaluated Algorithms/Models

This section describes the machine learning algorithms used in this study: RF, XGBoost, LightGBM, CatBoost, Gradient Boosting, Histogram Gradient Boosting, AdaBoost, Extra Trees, Decision Tree, Bagging, Linear Regression, Ridge, Lasso, Elastic Net, Huber, KNN, SVR and MLP. These algorithms are widely used in the literature.

2.3.1. Linear Models

Linear regression is a parametric statistical method that models the linear relationship between a dependent variable and one or more independent variables [35]. LASSO is a method that performs both regression and variable selection simultaneously while addressing the problem of multicollinearity. LASSO shrinks the coefficients toward zero by adding an L1 penalty term to OLS regression. This results in simpler and more interpretable models [36]. Ridge regression is a regularization method used to reduce the problem of multicollinearity found in multiple linear regression models and to improve the model's generalizability. Ridge regression reduces the coefficients by adding an L2 penalty term to the Ordinary Least Squares (OLS) cost function, thereby yielding more robust estimates. Ridge regression does not set the coefficients to zero entirely; it merely reduces them. Therefore, all variables remain in the model. This feature distinguishes it from the LASSO method [37]. Elastic Net regression is a regularization technique that combines Lasso regression and Ridge regression. It reduces the number of unnecessary variables through variable selection while also controlling for multicollinearity [38].

Huber regression [39] is a robust regression method developed to be more resistant to outliers and is used as an alternative to the traditional OLS method in cases of heavy-tailed error distributions.

2.3.2. Distance, Kernel and Neural-Network Models

In KNN regression, the distances between a new observation and the training observations are calculated using a selected distance metric. The k nearest observations are identified, and the continuous prediction is obtained from their target values, typically using an unweighted or distance-weighted average. Model performance depends on the choice of k , distance metric, weighting scheme, and feature scaling [40]. SVR estimates a continuous response by fitting a function within an epsilon-insensitive margin while controlling model complexity through regularization and kernel parameters [41]. MLP is a feedforward artificial neural network composed of interconnected layers of neurons whose weights are optimized during training [42].

2.3.3. Boosting Models

CatBoost is a gradient-boosted decision-tree algorithm. LightGBM and XGBoost are efficient gradient-boosting frameworks used for regression and classification. In this study, all categorical predictors were preprocessed through the common pipeline before model fitting. AdaBoost regression combines weak regressors sequentially and assigns greater emphasis to observations with larger prediction errors. Gradient Boosting regression builds successive weak learners to reduce residual prediction errors. Histogram Gradient Boosting improves computational efficiency by grouping continuous feature values into discrete bins [43–49].

2.3.4. Bagging and Tree Models

Decision trees are a machine learning algorithm that classifies data and makes predictions based on specified rules. Their basic components are the root node, decision nodes, branches, and leaf nodes [50]. RF is a machine learning algorithm widely used in both classification and regression problems. First developed by Breiman [51] in 2001, it provides high accuracy and generalizability performance on high-dimensional and complex datasets. RF is an ensemble learning method in which a large number of decision trees are trained using bootstrap sampling and random feature selection [52]. ET is an algorithm that combines the predictions of multiple decision trees. Unlike RF, it selects random thresholds at each node instead of the best threshold and builds each tree on the entire training dataset. In regression, predictions are obtained by averaging the tree outputs [53]. Bagging is a method used in both classification and regression problems. It reduces overfitting, lowers the model's variance, and improves its accuracy. In this algorithm, multiple decision trees are created from the dataset using bootstrap sampling. Each tree is trained independently, and the final prediction is obtained by taking the majority vote for classification and the average for regression [54].

2.4. Cross-Validation Data Partitioning

In this study, a no fixed training–holdout split was created. Generalization performance was estimated using repeated five-fold nested cross-validation. In each outer iteration, approximately 80% of the observations formed the outer-training partition and approximately 20% formed the outer-test fold. These partitions changed across folds and repetitions.

In this stage, we checked for exact and near-duplicate rows prior to cross-validation, since replicated scenarios would leak between train and test folds under random K-fold

partitioning. No exact duplicate rows were found, and no rows shared identical values across all predictors; near-duplicate overlap on the two dominant continuous predictors (temperature, solar radiation) was limited to 0.85% of rows, consistent with ordinary sampling density rather than with repeated scenarios. Random K-fold partitioning was therefore not expected to leak information via row duplication in this dataset.

Generalizability

In this study, the assumption that the observations were independently and identically distributed could not be verified due to the structure of the dataset and the lack of sufficient information provided by the data provider regarding possible dependency relationships between samples. Therefore, the applied row-based random cross-validation was considered not as a verified physical assumption that the observations in the dataset were independent but as a statistical evaluation strategy used to assess out-of-sample estimation under the current data structure.

Consequently, the obtained performance results were interpreted not as generalizability to independent field observations obtained from real buildings but as estimation performance on new observations from the same digital twin data distribution representing the same data generation process.

2.5. Nested Cross-Validation

Nested cross-validation was employed to obtain a robust and unbiased assessment of the generalizability of the machine learning models, reduce prediction bias, and minimize overfitting. The procedure consisted of an outer loop for model performance evaluation and an inner loop for hyperparameter optimization. Figure 3 illustrates the nested five-fold cross-validation framework used in this study.

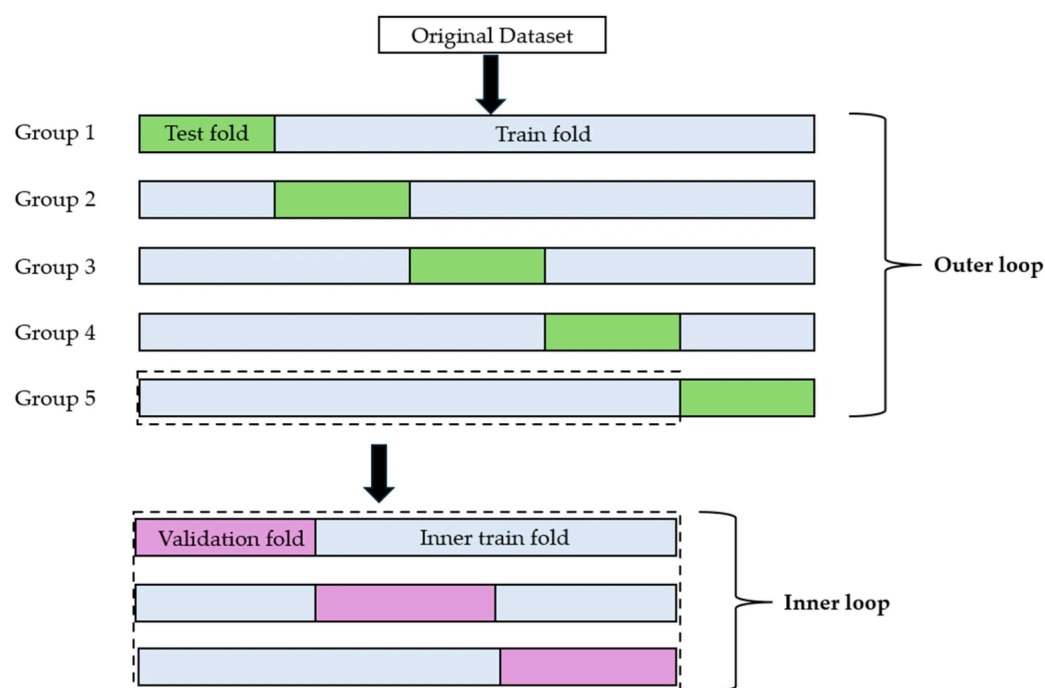


Figure 3. Repeated five-fold nested cross-validation; the outer five-fold procedure was repeated ten times [55].

Nested cross-validation was applied separately to each machine learning model. In the outer loop, five-fold cross-validation was repeated ten times, producing a total of 50

outer-loop training and test splits. Within each outer-loop training set, an additional five-fold cross-validation procedure was conducted to optimize the model hyperparameters.

2.5.1. Outer Loop: Performance Evaluation

For each model, the complete analysis dataset was evaluated using repeated five-fold outer cross-validation with ten repetitions. In each repetition, the dataset was randomly divided into five approximately equal folds. Four folds, corresponding to approximately 80% of the observations, formed the outer-training partition, while the remaining fold, corresponding to approximately 20%, formed the outer-test fold.

Each fold served once as the test set within a five-fold cycle. The complete five-fold procedure was repeated ten times using different random partitions, resulting in 50 outer-loop training and test splits. The test set in each outer-loop iteration remained completely separate from the hyperparameter optimization and model-fitting procedures. Consequently, each model was evaluated using observations that had not been used during either model training or hyperparameter selection.

The performance metrics obtained from the 50 outer-loop evaluations were subsequently summarized to estimate the generalization performance of each model.

2.5.2. Inner Loop: Hyperparameter Optimization

Hyperparameter optimization was performed using Optuna (4.9.0), an open-source Python framework developed by Akiba et al. [56]. In this study, Optuna used the Tree-structured Parzen Estimator sampler to propose hyperparameter configurations from predefined search spaces. For each outer-loop iteration, optimization was conducted exclusively within the corresponding outer-training partition using inner five-fold cross-validation. The outer-test fold was not used during tuning.

2.5.3. Model Retraining and Outer-Loop Evaluation

After the optimal hyperparameters had been identified in the inner loop, the corresponding model was retrained using the complete outer-loop training set. The retrained model was then used to predict the responses of the outer-loop test set. For each outer-loop test set, model performance was evaluated using the coefficient of determination (R^2), root mean square error (RMSE), mean absolute error (MAE), and mean absolute percentage error (MAPE). Because the outer-loop test observations were excluded from both hyperparameter optimization and model training, these metrics represented out-of-sample performance estimates. This procedure was repeated for all 50 outer-loop splits and for each of the 18 machine learning models. Consequently, 50 values of each performance metric were obtained for every model. The mean and standard deviation of these values were calculated to summarize model performance across the repeated outer-loop evaluations.

For each model and performance metric, a 95% percentile-bootstrap confidence interval was calculated for the mean outer-fold score. The 50 outer-fold metric values were resampled with replacement 10,000 times, the mean was calculated for each resample, and the 2.5th and 97.5th percentiles of the bootstrap distribution are reported. Because repeated cross-validation produces overlapping partitions, these intervals should be interpreted as descriptive uncertainty intervals over the observed outer-fold scores.

The models were ranked according to their mean nested cross-validation RMSE values. The model with the lowest mean RMSE was identified as the best-performing model. Finally, the selected model was retrained using the entire dataset, and permutation importance analysis was applied to quantify the influence of each predictor variable

on the model predictions.

2.6. Model Evaluation Metrics

To evaluate model performance in a more comprehensive and reliable manner, it is recommended to use multiple performance metrics together. In this study, the performance of the classical machine learning models was evaluated without relying on a single metric. RMSE was selected as the primary criterion for the hyperparameter optimization process; Root Mean Square Error (RMSE), Mean Absolute Error (MAE), Coefficient of Determination (R^2), and Mean Absolute Percentage Error (MAPE) were selected to report the final model performance. After the summaries of all models, graphs and detailed metric values for the best machine learning model are presented.

2.6.1. Root Mean Square Error (RMSE)

RMSE is calculated as the square root of the mean of the squared differences between predicted and observed values (Equation (1)). A lower RMSE indicates better predictive performance.

$$\text{RMSE} = \sqrt{\frac{1}{N} \sum_{i=1}^N (y_{\text{actual}} - y_{\text{predicted}})^2} \quad (1)$$

2.6.2. Mean Absolute Error (MAE)

MAE measures the average of the absolute differences between the predicted values and the actual values (Equation (2)) and indicates the magnitude of the model's prediction error [57].

$$\text{MAE} = \frac{1}{N} \sum_{i=1}^N |y_{\text{actual}} - y_{\text{predicted}}| \quad (2)$$

2.6.3. Coefficient of Determination (R^2)

R^2 measures how well the prediction model fits the given data (Equation (3)). It indicates how much of the variation in the actual results is captured by the model [58].

$$R^2 = 1 - \frac{\sum_{i=1}^N (y_{\text{actual}} - y_{\text{predicted}})^2}{\sum_{i=1}^N (y_{\text{actual}} - y_{\text{average}})^2} \quad (3)$$

2.6.4. Mean Absolute Percentage Error (MAPE)

MAPE is a performance metric that indicates the extent to which estimated values deviate from actual values (Equation (4)). MAPE expresses the prediction error as a percentage, making it easier to compare and interpret results across different scales [58].

$$\text{MAPE} = \frac{1}{N} \sum_{i=1}^N \left(\frac{|y_{\text{actual}} - y_{\text{predicted}}|}{y_{\text{actual}}} \right) \times 100 \quad (4)$$

In Equations (1)–(4), $y_{\text{predicted}}$, y_{actual} , and N represent the predicted value, the actual value, and the sample size, respectively.

2.7. Computational Environment and Reproducibility

Table 4 presents the computational environment of the study and the basic settings used for experimental reproducibility. The table includes the Python version used, operating system and hardware information, and the versions of the basic Python packages.

It also provides the random seeds used in the Nested Cross-Validation structure, the number of outer and inner CV layers, the number of Optuna trials, the bootstrap seed, and the parallel execution setting of the model. This information is provided to enable the reproduction of the experimental results in the same or a similar computational environment.

Table 4. Computational environment and experimental settings.

Feature	Description
python_version	3.12.4
platform	Windows-11-10.0.26200-SP0
processor	Intel 64 Family 6 Model 165 Stepping 5, GenuineIntel
machine	AMD64
cpu_count_logical	16
<i>package_versions</i>	
numpy [59]	1.26.3
pandas [60]	2.2.2
scikit-learn [61]	1.6.1
optuna [56]	4.9.0
xgboost [46]	3.0.3
lightgbm [62]	4.7.0
catboost [43]	1.2.10
scipy [63]	1.14.0
<i>seeds</i>	
CV_RANDOM_STATE	0
OUTER_CV_FOLDS	5
OUTER_CV_REPEATS	10
INNER_CV_FOLDS	5
N_TRIALS	50
BOOTSTRAP_SEED	0
note	inner Optuna study seed per outer fold = CV_RANDOM_STATE + fold_idx
model_n_jobs	1

A full Nested-CV optimization, including 18 models, 50 outer layers, and 50 Optuna attempts for each inner optimization, has a computation time of 32,546.5 s (approximately 9 h).

3. Results

All performance results presented in this section are based on 50 outer-fold evaluations obtained through repeated five-fold nested cross-validation with ten repetitions. Model performance was summarized using the mean $\pm$ standard deviation of R^2 , RMSE, MAE, and MAPE, together with 95% percentile-bootstrap confidence intervals for the mean outer-fold scores. Hyperparameter optimization was performed only within the inner folds; outer-test folds were not involved in model selection or tuning. These results provide estimates of model generalization performance within the evaluated simulation-generated distribution.

All codes and result sets related to this study are available at the GitHub repository [64].

Table 5 compares the predictive performance of all the machine learning models used.

Table 5. Performance comparison of machine learning models under nested cross-validation (mean $\pm$ standard deviation).

Model	R ² (Mean $\pm$ SD)	RMSE (kWh) (Mean $\pm$ SD)	MAE (kWh) (Mean $\pm$ SD)	MAPE (Proportion)	Runtime Seconds
CatBoost	0.8275 $\pm$ 0.0072	1.6792 $\pm$ 0.021	1.4356 $\pm$ 0.0233	0.0521 $\pm$ 0.0009	8621.9
XGBoost	0.8262 $\pm$ 0.0072	1.6858 $\pm$ 0.022	1.4364 $\pm$ 0.0236	0.0521 $\pm$ 0.0009	1353.1
GradientBoosting	0.8257 $\pm$ 0.0073	1.688 $\pm$ 0.0218	1.439 $\pm$ 0.0237	0.0522 $\pm$ 0.0009	5021.7s
LightGBM	0.8249 $\pm$ 0.0078	1.6918 $\pm$ 0.0236	1.4403 $\pm$ 0.0249	0.0523 $\pm$ 0.001	748.2
HistGradientBoosting	0.8239 $\pm$ 0.0077	1.6968 $\pm$ 0.024	1.4449 $\pm$ 0.0251	0.0524 $\pm$ 0.001	1569.7
RandomForest	0.8232 $\pm$ 0.0072	1.7001 $\pm$ 0.0223	1.4419 $\pm$ 0.0231	0.0523 $\pm$ 0.0009	0.4
ExtraTrees	0.823 $\pm$ 0.0071	1.7012 $\pm$ 0.0231	1.4453 $\pm$ 0.0238	0.0525 $\pm$ 0.0009	3301.4689
Lasso	0.8195 $\pm$ 0.0079	1.7176 $\pm$ 0.0231	1.4845 $\pm$ 0.026	0.0538 $\pm$ 0.001	92.4329
ElasticNet	0.8195 $\pm$ 0.0079	1.7176 $\pm$ 0.0231	1.4845 $\pm$ 0.026	0.0538 $\pm$ 0.001	99.4973
LinearRegression	0.8194 $\pm$ 0.008	1.7182 $\pm$ 0.0235	1.4849 $\pm$ 0.0263	0.0538 $\pm$ 0.001	9.5269
HuberRegressor	0.8194 $\pm$ 0.008	1.7182 $\pm$ 0.0235	1.4849 $\pm$ 0.0263	0.0538 $\pm$ 0.001	118.9971
Ridge	0.8194 $\pm$ 0.008	1.7182 $\pm$ 0.0235	1.4848 $\pm$ 0.0263	0.0538 $\pm$ 0.001	86.2477
AdaBoost	0.8191 $\pm$ 0.0073	1.7198 $\pm$ 0.0201	1.4793 $\pm$ 0.0228	0.0535 $\pm$ 0.0009	0.4
Bagging	0.8176 $\pm$ 0.0071	1.7271 $\pm$ 0.0243	1.4589 $\pm$ 0.0242	0.053 $\pm$ 0.001	6573.0806
MLP	0.811 $\pm$ 0.0118	1.7575 $\pm$ 0.0418	1.4836 $\pm$ 0.0348	0.0538 $\pm$ 0.0013	15,475.5447
SVR	0.8058 $\pm$ 0.0084	1.7817 $\pm$ 0.026	1.4971 $\pm$ 0.022	0.0544 $\pm$ 0.0008	5920.7727
DecisionTree	0.7938 $\pm$ 0.0084	1.8363 $\pm$ 0.0271	1.5267 $\pm$ 0.0257	0.0553 $\pm$ 0.001	140.8673
KNN	0.7227 $\pm$ 0.0069	2.13 $\pm$ 0.0306	1.7257 $\pm$ 0.0289	0.0635 $\pm$ 0.0011	463.9847

CatBoost was selected as the final (best) model after achieving the lowest RMSE and MAE values among all models with full numerical precision. CatBoost's ability to successfully capture complex nonlinear relationships and interactions among features contributed to its high prediction accuracy. Although XGBoost also demonstrated strong performance, CatBoost was chosen as the final model due to its lower RMSE value and higher overall prediction accuracy. KNN achieved the highest mean RMSE. Under the evaluated preprocessing and hyperparameter settings, KNN was less effective than the linear and ensemble models in approximating the simulation-generated input-output relationship.

The final model selection was based not only on the lowest raw RMSE value but also on the 1-SE prudence rule, which takes into account the balance between prediction performance and model complexity.

Table 6 shows the 95% confidence intervals for the nested cross-validation performance metrics for all models. The corresponding 95% bootstrap confidence intervals (95% CI) for CatBoost are 0.8255–0.8295 for R², 1.6733–1.6851 kWh for RMSE, 1.6251–1.6383 kWh for MAE, and 0.0518–0.0523 for MAPE.

Table 6. The 95% confidence intervals of the nested cross-validation performance metrics for all models (low to high).

Model	R ² (95% CI)	RMSE (95% CI)	MAE (95% CI)	MAPE (95% CI)
CatBoost	0.8255–0.8295	1.6733–1.6851	1.6251–1.6383	0.0518–0.0523
XGBoost	0.8241–0.8282	1.6796–1.6919	1.6252–1.6383	0.0518–0.0524
GradientBoosting	0.8237–0.8277	1.6817–1.694	1.6261–1.6392	0.0519–0.0525
LightGBM	0.8228–0.8271	1.6851–1.6983	1.6261–1.6393	0.052–0.0525
HistGradientBoosting	0.8217–0.826	1.6899–1.7035	1.6261–1.6393	0.0521–0.0527
RandomForest	0.8212–0.8252	1.6939–1.7063	1.6299–1.6431	0.0521–0.0526
ExtraTrees	0.821–0.8249	1.6947–1.7075	1.6334–1.6469	0.0522–0.0527
Lasso	0.8174–0.8217	1.7111–1.724	1.6344–1.6476	0.0535–0.054
ElasticNet	0.8174–0.8217	1.7111–1.724	1.6353–1.6483	0.0535–0.054
LinearRegression	0.8172–0.8217	1.7115–1.7247	1.6365–1.6497	0.0535–0.054
HuberRegressor	0.8172–0.8217	1.7115–1.7247	1.638–1.6515	0.0535–0.054
Ridge	0.8172–0.8217	1.7115–1.7247	1.6486–1.6624	0.0535–0.054
AdaBoost	0.8171–0.8211	1.714–1.7251	1.658–1.6703	0.0533–0.0537
Bagging	0.8156–0.8195	1.7203–1.7337	1.6653–1.6805	0.0528–0.0533
MLP	0.8075–0.814	1.7469–1.7701	1.674–1.6903	0.0534–0.0542
SVR	0.8035–0.8081	1.7746–1.7891	1.6899–1.7044	0.0541–0.0546
DecisionTree	0.7914–0.796	1.8288–1.8438	1.7048–1.7194	0.0551–0.0556
KNN	0.7207–0.7245	2.1215–2.1384	1.8703–1.8891	0.0632–0.0639

Figure 4 shows the RMSE differences obtained for 18 tuned models under Nested CV, using CatBoost as the reference model. The points shown in the graph appear to represent the summarized difference per model plus the 95% paired bootstrap CI. The fact that the Δ RMSE values are positive for all models indicates that CatBoost has the lowest RMSE value among the evaluated models. While performance closest to CatBoost is observed in the XGBoost and Gradient Boosting models, the RMSE difference is noticeably higher in the KNN, Decision Tree, and SVR models. These results show that CatBoost not only achieves the lowest average RMSE but also provides a significant error advantage compared to alternative models.

To improve readability, only summary performance results are provided in the main text, not detailed result tables. Mean paired differences, Cohen's d_z effect sizes, paired t -test and Wilcoxon signed-rank test results, and Holm-corrected p -values for pairwise statistical comparisons of tuned models in terms of RMSE are presented in Supplementary Table S1.

Optuna's convergence behavior during hyperparameter optimization is shown in Table 7. Optuna's convergence results indicate that hyperparameter searches stabilized at different rates across models. Both LightGBM and XGBoost used nine hyperparameters. While XGBoost reached the 1% range of its final performance earlier (median of 6.5 trials), LightGBM reached this level after 10.5 trials. Since there were no hyperparameters to optimize for LinearRegression, a convergence analysis could not be performed. CatBoost reached the 1% range of its final performance with five hyperparameters in a median of 3 trials and showed an average additional improvement of 1.83% in the bottom 25% of trials. Convergence speed was not evaluated as a standalone indicator of model performance. The final model selection was based on Nested-CV RMSE results.

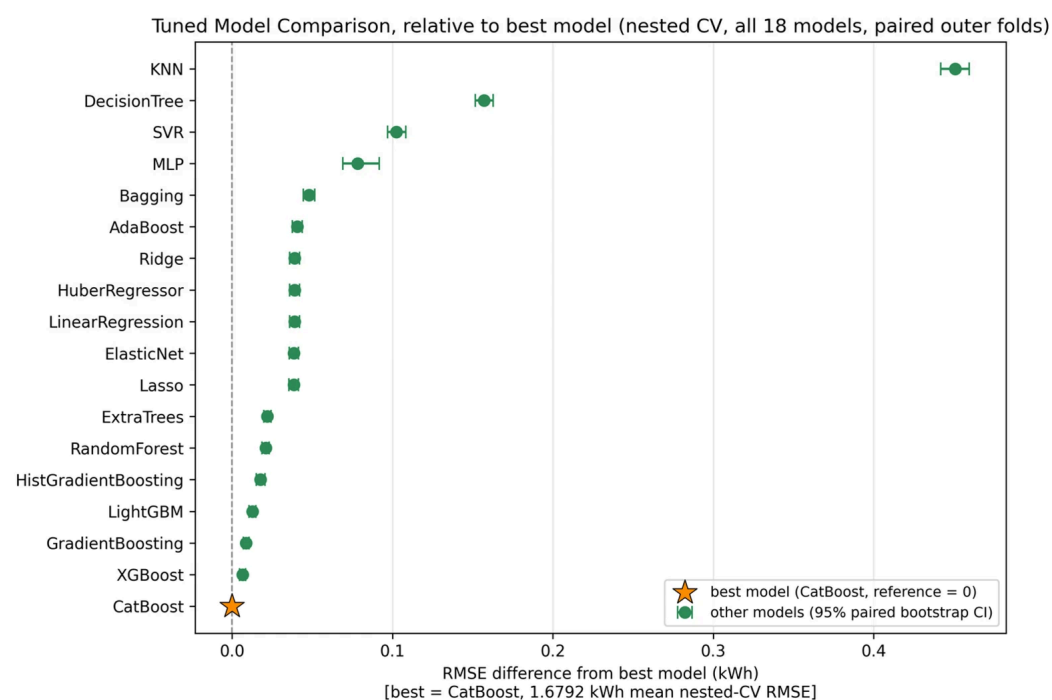


Figure 4. Tuned model comparison, relative to best model (Nested CV, error bars = 95% bootstrap CI).

Table 7. Optuna hyperparameter optimization convergence summary.

Model	No. of Hyperparameters	Median Trial to Within 1% of Final Performance	Folds Still Improving in Final 25% of Trials (%)	Mean Improvement in Final 25% (%)
LightGBM	9	10.5	4	1.2953
XGBoost	9	6.5	2	0.5872
HistGradientBoosting	6	7	6	1.2946
CatBoost	5	3	8	1.8297
GradientBoosting	5	4	8	2.9538
RandomForest	5	1	18	3.8289
ExtraTrees	5	1.5	16	2.7362
Bagging	3	8.5	18	4.1675
DecisionTree	3	7.5	20	6.4791
SVR	3	13.5	16	3.7631
MLP	3	8	22	3.0035
KNN	3	2	2	0.4775
AdaBoost	2	7.5	4	1.1190
HuberRegressor	2	1	14	2.6754
ElasticNet	2	1	24	4.4009
Ridge	1	1	0	0.1154
Lasso	1	1	0	0.1595
LinearRegression	-	-	-	-

CatBoost achieved the lowest RMSE value among all evaluated models and was therefore used as the reference model. As shown in Table 8, XGBoost demonstrated performance closest to that of CatBoost (Δ RMSE = 0.0066 kWh, 95% CI: 0.0046–0.0085 kWh).

This was followed by the Gradient Boosting (0.0088 kWh) and LightGBM (0.0126 kWh) models. The KNN model, on the other hand, showed the largest performance difference compared to CatBoost, with an RMSE difference of 0.4509 kWh (95% CI: 0.4418–0.4596 kWh).

Table 8. RMSE differences of the tuned models relative to the best-performing CatBoost model.

Model	Mean Diff	CI (Low)	CI (High)
CatBoost	0	0	0
XGBoost	0.0066	0.0046	0.0085
GradientBoosting	0.0088	0.0071	0.0105
LightGBM	0.0126	0.0105	0.0148
HistGradientBoosting	0.0176	0.0152	0.0205
RandomForest	0.0209	0.0186	0.0232
ExtraTrees	0.0221	0.0198	0.0244
Lasso	0.0384	0.0355	0.0414
ElasticNet	0.0385	0.0356	0.0414
LinearRegression	0.039	0.0359	0.0421
HuberRegressor	0.039	0.0359	0.0421
Ridge	0.039	0.0359	0.0421
AdaBoost	0.0406	0.0375	0.0438
Bagging	0.0479	0.0444	0.0515
MLP	0.0784	0.069	0.0918
SVR	0.1026	0.0969	0.1085
DecisionTree	0.1572	0.1517	0.1627
KNN	0.4509	0.4418	0.4596

3.1. Model Selection and Generalization Performance

All candidate models were subjected to hyperparameter optimization under the same Nested Cross-Validation framework, and their performance was compared based on out-of-bag test results. The final model selection was not based solely on the raw minimum of a single performance metric. Prediction performance, model complexity, and generalization capacity were evaluated together. In this context, the final model was selected from among simpler models that demonstrated performance close to the lowest RMSE value, using a one-standard-error (1-SE) conservatism rule.

As a result of this evaluation, CatBoost was selected as the final model. CatBoost was trained with five tuned hyperparameters and ranked among the best solutions in terms of performance, with its raw minimum RMSE falling within one standard error. However, the key factor in the model selection was the low and consistent prediction error achieved by CatBoost in the outer layers of Nested-CV.

For CatBoost, the Nested-CV RMSE value obtained was 1.6792 ± 0.0210 kWh, and the 95% bootstrap confidence interval calculated using the outer folds was 1.6733–1.6851 kWh. In contrast, the MAE was found to be 1.4356 ± 0.0233 kWh, R^2 was 0.8275 ± 0.0072 , and MAPE was 0.0521 ± 0.0009 . These results indicate that the model exhibits similar performance across different outer layers and that its generalization performance shows low variability.

The selection of CatBoost was also supported by paired RMSE comparisons with other tuned models. When CatBoost was used as the reference, the RMSE difference for XGBoost—the second-best model—was only 0.0066 kWh (95% CI: 0.0046–0.0085 kWh),

while for Gradient Boosting and LightGBM, these differences were 0.0088 and 0.0126 kWh, respectively. The difference increased progressively for other models, reaching 0.4509 kWh for KNN. CatBoost achieved the lowest RMSE and, when compared to other models—particularly XGBoost—demonstrated very similar performance.

The final model selection was not based solely on the raw minimum RMSE. The 1-SE conservatism rule ensured that a simpler model was preferred among solutions that performed statistically close to the lowest RMSE. CatBoost's structure, which includes only five hyperparameters, and the fact that its raw minimum performance remained within a standard error threshold, demonstrate that the model offers an appropriate balance between performance and model complexity.

The results of the Breusch–Pagan test indicated the presence of heteroscedasticity ($p < 0.001$), and the Shapiro–Wilk test revealed a deviation from normality ($p < 0.001$). These findings point to a limitation related to the distributional properties of the residuals rather than the model's predictive performance. Therefore, CatBoost's generalization performance was evaluated using Nested CV outer-layer results and bootstrap confidence intervals, independent of classical residual assumptions.

Following the evaluation of comparative model performance, a detailed analysis was conducted only on the models that exhibited the highest performance. Accordingly, the results for the CatBoost model, which yielded the lowest RMSE value, are presented below.

Table 9 summarizes the outer-fold performance of the CatBoost model. The model explained approximately 82.8% of the variance in the target variable, with $R^2 = 0.8275 \pm 0.0072$. RMSE was 1.6791 ± 0.0209 kWh, MAE was 1.4355 ± 0.0233 kWh, and MAPE was 0.0520 ± 0.0008 . The relatively narrow distribution of the outer-fold scores indicates limited variation across the repeated partitions of this simulation-generated dataset.

Table 9. Generalization estimate = nested CV outer-fold scores (mean $\pm$ std [95% bootstrap CI over outer folds]) for best-tuned model (CatBoost). one_se_rule_candidates.csv.

Metric	Nested CV Mean	Nested CV Std	CI95 (Low)	CI95 (High)
R^2	0.8275	0.0072	0.8255	0.8295
RMSE	1.6791	0.0209	1.6732	1.6850
MAE	1.4355	0.0233	1.4290	1.4420
MAPE	0.0520	0.0008	0.0518	0.0523

Table 10 shows the per-fold metrics of the selected CatBoost model across 50 outer folds. The model demonstrated relatively consistent prediction performance across the outer folds. The relatively low standard deviations between outer folds indicate that variability in model performance across different data splits is limited and that the generalization performance is stable.

Table 10. Per-fold metrics of the selected CatBoost model across 50 outer folds of Nested CV.

Fold	Outer R^2	Outer RMSE	Outer MAE	Outer MAPE
0	0.8238	1.6995	1.4573	0.0528
1	0.8357	1.6471	1.3967	0.0508
2	0.812	1.7122	1.4775	0.0537
3	0.8303	1.6866	1.4443	0.0528
4	0.8314	1.6693	1.417	0.0509
5	0.8337	1.6404	1.399	0.0506

Fold	Outer R ²	Outer RMSE	Outer MAE	Outer MAPE
6	0.8141	1.6762	1.4334	0.0518
7	0.8276	1.6688	1.423	0.0514
8	0.8332	1.6984	1.4522	0.0531
9	0.8302	1.704	1.4566	0.0531
10	0.824	1.711	1.4677	0.053
11	0.8226	1.6825	1.4449	0.0522
12	0.823	1.6751	1.4343	0.0522
13	0.8335	1.657	1.4075	0.0511
14	0.837	1.6621	1.4207	0.0518
15	0.8221	1.6728	1.43	0.0519
16	0.8247	1.7147	1.4785	0.0534
17	0.8267	1.6987	1.4531	0.0528
18	0.8386	1.6467	1.4077	0.051
19	0.8288	1.6498	1.3968	0.0509
20	0.8123	1.7076	1.4603	0.0529
21	0.8352	1.6612	1.4096	0.0511
22	0.8344	1.6669	1.4163	0.0515
23	0.8339	1.6869	1.4532	0.0527
24	0.8166	1.6955	1.4572	0.0529
25	0.8377	1.6639	1.4118	0.0507
26	0.813	1.6923	1.4353	0.0522
27	0.8225	1.687	1.4638	0.053
28	0.8371	1.6529	1.4068	0.0515
29	0.8269	1.6958	1.4562	0.0528
30	0.828	1.6495	1.4061	0.0508
31	0.8308	1.6731	1.4218	0.0516
32	0.8287	1.6773	1.4355	0.0523
33	0.8283	1.6727	1.44	0.0522
34	0.8253	1.7125	1.4742	0.0534
35	0.8269	1.6932	1.4502	0.0525
36	0.8321	1.6546	1.4145	0.0511
37	0.8285	1.6874	1.4375	0.0525
38	0.8266	1.6536	1.414	0.0512
39	0.827	1.6962	1.4579	0.0528
40	0.8222	1.7075	1.4629	0.0534
41	0.8275	1.6481	1.4142	0.0509
42	0.8297	1.6675	1.4215	0.0513
43	0.837	1.6694	1.4306	0.0523
44	0.8227	1.6988	1.4456	0.0524
45	0.8431	1.6465	1.394	0.0504
46	0.8225	1.702	1.4634	0.0529
47	0.8334	1.6808	1.418	0.052
48	0.8137	1.6956	1.4501	0.0528
49	0.8202	1.689	1.4569	0.0524

Table 11 shows the CatBoost hyperparameters selected based on the results of the inner CV + Optuna for each of the 50 outer folds. This table illustrates how much the hyperparameters vary across the outer folds and demonstrates the stability of the optimization.

Table 11. Per-fold best hyperparameters selected for CatBoost during Nested CV.

Fold	Seed	Inner CV RMSE	Iterations	Depth	Learning_Rate	Min_Leaf_Weight	Subsampling_T
0	0	1.6789	450	4	0.0203	5.2114	0.0234
1	1	1.6918	450	5	0.0193	6.4443	0.796
2	2	1.6733	600	5	0.0157	4.5453	0.1589
3	3	1.6778	450	5	0.0217	5.3271	0.0809
4	4	1.6839	600	6	0.0175	6.6226	0.7263
5	5	1.6937	650	6	0.0178	2.1684	0.108
6	6	1.6854	400	6	0.0273	8.5843	0.1571
7	7	1.685	750	5	0.0132	5.9702	0.5089
8	8	1.6806	400	5	0.0273	3.339	0.1473
9	9	1.6787	550	6	0.0153	4.92	0.8704
10	10	1.67	800	4	0.0139	8.7049	0.8241
11	11	1.6837	600	5	0.0166	2.6763	0.5308
12	12	1.6825	700	5	0.0151	2.7494	0.331
13	13	1.6925	300	6	0.0347	5.3812	0.7732
14	14	1.6858	700	7	0.0135	5.8186	0.755
15	15	1.6862	750	5	0.0177	7.1065	0.5698
16	16	1.6662	800	6	0.0112	1.1207	0.5471
17	17	1.6806	650	4	0.0148	4.834	0.0861
18	18	1.6922	550	6	0.0203	6.7319	0.9818
19	19	1.6858	300	5	0.0322	5.6269	0.8824
20	20	1.6736	650	5	0.0176	4.9163	0.8436
21	21	1.6893	600	5	0.017	6.2318	0.9595
22	22	1.6812	500	7	0.0206	1.091	0.5847
23	23	1.6857	300	5	0.0279	6.9193	0.0797
24	24	1.6755	800	5	0.0159	7.9932	0.8402
25	25	1.6845	750	5	0.0141	2.799	0.6174
26	26	1.6759	600	5	0.0195	2.5392	0.8667
27	27	1.6751	650	6	0.0135	3.1424	0.4491
28	28	1.6888	650	5	0.0125	1.1958	0.5311
29	29	1.6797	400	6	0.0242	6.9743	0.2696
30	30	1.6929	750	5	0.0153	4.7664	0.3523
31	31	1.6866	600	6	0.0139	7.077	0.0857
32	32	1.6837	700	5	0.0171	5.8144	0.0807
33	33	1.6831	700	4	0.0144	2.7025	0.9838
34	34	1.6722	700	6	0.0154	4.9937	0.9409
35	35	1.6748	550	6	0.0185	6.7529	0.6045
36	36	1.6902	750	5	0.0121	4.9218	0.2693
37	37	1.6736	400	5	0.0231	1.053	0.3763

Fold	Seed	Inner CV RMSE	Iterations	Depth	Learning_Rate	Min_Leaf_Weight	Subsampling_T
38	38	1.687	450	7	0.0238	4.2067	0.5788
39	39	1.679	650	5	0.0157	9.2425	0.9281
40	40	1.6704	400	5	0.0218	2.7183	0.9464
41	41	1.6878	450	6	0.02	3.3121	0.6976
42	42	1.6871	450	5	0.0201	3.7898	0.1646
43	43	1.6839	650	6	0.0123	4.3799	0.6691
44	44	1.6764	800	6	0.0115	2.0212	0.2179
45	45	1.6971	250	6	0.024	3.2676	0.1102
46	46	1.6757	550	5	0.018	9.407	0.9634
47	47	1.6828	250	5	0.0356	8.9697	0.7582
48	48	1.6763	600	6	0.019	4.9675	0.2447
49	49	1.6817	600	6	0.0165	1.2396	0.2992

Figure 5 shows the convergence behavior of Optuna-based hyperparameter optimization for CatBoost across 50 outer folds. The X-axis represents the number of Optuna trials, while the Y-axis represents the best inner CV RMSE value obtained up to that point. The fact that the running-best RMSE decreases in the initial trials and then levels off in subsequent trials indicates that the optimization identifies better hyperparameter configurations in the early stages but that the additional gains obtained in later trials gradually diminish. This suggests that the hyperparameter search stabilizes to a large extent before reaching the maximum of 50 trials.

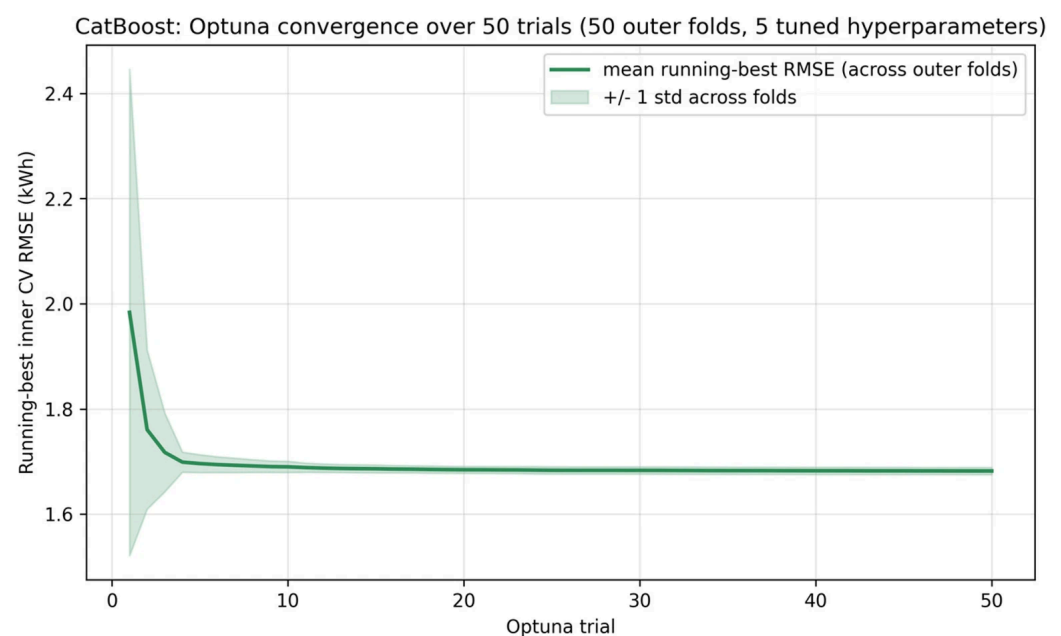


Figure 5. Convergence of the running-best inner-CV RMSE during CatBoost hyperparameter optimization across 50 outer folds.

The numerical convergence results for CatBoost hyperparameter optimization are presented in Table 12. The fact that the running-best RMSE becomes increasingly stable in subsequent trials is consistent with Figure 5. This convergence behavior indicates that Optuna identified competitive hyperparameter configurations in the early stages of the search process, while subsequent trials yielded only limited additional improvements.

Table 12. Optuna convergence of the CatBoost hyperparameter optimization across 50 outer folds.

Trial	Running Best RMSE (Mean)	Running Best RMSE (Std)
1	1.983669587	0.462185849
2	1.760725826	0.151003076
3	1.717631742	0.074973977
4	1.699031387	0.019120613
5	1.696188738	0.017027425
6	1.694236152	0.015052759
7	1.692887592	0.01374414
8	1.691705975	0.012150554
9	1.690499132	0.010895451
10	1.690024442	0.01071677
11	1.688576564	0.008783819
12	1.687591253	0.007999976
13	1.686907583	0.007602482
14	1.686567033	0.007645698
15	1.68627641	0.007532254
16	1.685751094	0.00709262
17	1.685510751	0.006880418
18	1.685078473	0.006932018
19	1.684686036	0.006635274
20	1.684416289	0.006569956
21	1.68436836	0.006597892
22	1.684159591	0.006643988
23	1.684040872	0.006698487
24	1.683708635	0.007000401
25	1.683405732	0.006898631
26	1.683341458	0.006938441
27	1.683341458	0.006938441
28	1.683341458	0.006938441
29	1.68330217	0.006957575
30	1.683300187	0.006957952
31	1.683281402	0.006975048
32	1.683093221	0.007052962
33	1.682997708	0.00711901
34	1.682856983	0.007011572
35	1.682813963	0.006977066
36	1.682802291	0.006975085
37	1.682776045	0.006957397
38	1.682673512	0.006936621
39	1.682611673	0.006925234
40	1.682548171	0.006955276
41	1.682548171	0.006955276
42	1.682521641	0.006968638
43	1.682400687	0.006884846

Trial	Running Best RMSE (Mean)	Running Best RMSE (Std)
44	1.682400106	0.006885702
45	1.682397535	0.006882761
46	1.682256156	0.00673979
47	1.682252683	0.006744222
48	1.682192387	0.006772087
49	1.682128805	0.006866928
50	1.682120844	0.00685501

Overall, the convergence curve and trial-based statistics support the conclusion that the CatBoost hyperparameter search stabilized to a large extent before reaching the maximum of 50 trials and that the optimization budget used was sufficient.

The convergence characteristics of the Optuna search on a per-fold basis are presented in Table 13. Fifty trials were allocated to all outer folds, and the convergence rate varied across folds. While some folds reached the final performance region within the first few trials, others required more optimization to approach the final inner-CV RMSE value. The fact that the improvement observed in the top 25% of trials varied across folds also indicates that the optimization surface is not the same across different outer training subsets. These results support the use of a sufficient trial budget to account for fold-specific differences in hyperparameter search.

Table 13. Optuna convergence per fold.

Fold	Seed	No. of Trials	First Trial RMSE	Final RMSE	Trial Within 1% of Final	Improvement in Final 25% (%)
0	0	50	1.6955	1.6789	1	14.6337
1	1	50	3.1325	1.6918	4	0.0756
2	2	50	1.6825	1.6733	1	15.4084
3	3	50	1.7742	1.6778	4	0.4005
4	4	50	1.8393	1.6839	11	0
5	5	50	2.6546	1.6937	8	0.1593
6	6	50	1.7676	1.6854	2	1.1346
7	7	50	1.9716	1.685	3	0.6976
8	8	50	1.7903	1.6806	12	0
9	9	50	1.9231	1.6787	9	0
10	10	50	1.6804	1.67	1	0
11	11	50	1.8197	1.6837	2	2.3083
12	12	50	2.4819	1.6825	4	0
13	13	50	1.743	1.6925	4	0.0598
14	14	50	1.7955	1.6858	2	0
15	15	50	2.4388	1.6862	6	0.1583
16	16	50	1.6727	1.6662	1	0
17	17	50	2.4132	1.6806	4	0
18	18	50	1.7998	1.6922	8	0.37
19	19	50	2.795	1.6858	3	0
20	20	50	1.7703	1.6736	2	0.1795
21	21	50	1.7013	1.6893	1	35.2698
22	22	50	1.7651	1.6812	2	0
23	23	50	1.7688	1.6857	2	0

Fold	Seed	No. of Trials	First Trial RMSE	Final RMSE	Trial Within 1% of Final	Improvement in Final 25% (%)
24	24	50	1.8602	1.6755	3	0
25	25	50	1.7128	1.6845	4	0
26	26	50	1.7023	1.6759	11	2.1267
27	27	50	1.7124	1.6751	5	0
28	28	50	2.1818	1.6888	12	0.1636
29	29	50	2.3088	1.6797	4	0
30	30	50	1.7205	1.6929	7	0.0546
31	31	50	1.7437	1.6866	3	0
32	32	50	1.6888	1.6837	1	0
33	33	50	1.7651	1.6831	5	0
34	34	50	3.63	1.6722	5	0.02
35	35	50	2.0342	1.6748	17	0
36	36	50	1.8572	1.6902	5	0.599
37	37	50	1.7914	1.6736	3	0.7523
38	38	50	1.7937	1.687	3	0.7361
39	39	50	1.7618	1.679	2	0
40	40	50	1.6985	1.6704	2	11.3092
41	41	50	1.6963	1.6878	1	0
42	42	50	1.7352	1.6871	6	0.9948
43	43	50	3.255	1.6839	11	0.1376
44	44	50	1.7091	1.6764	2	0
45	45	50	1.7113	1.6971	1	0
46	46	50	1.7795	1.6757	2	3.1603
47	47	50	1.7162	1.6828	2	0
48	48	50	3.0114	1.6763	4	0
49	49	50	1.7296	1.6817	3	0.5771

Table 14 summarizes the convergence behavior of CatBoost hyperparameter optimization across 50 outer folds. The number of hyperparameters indicates the complexity of the optimized model. The average and median number of trials required to reach 1% deviation from the final performance shows how early the optimization converged. The percentage of folds showing improvement in the last 25% of trials and the average improvement during this period assess whether there is still a significant gain in the final stage of optimization.

Table 14. Summary of the convergence characteristics of CatBoost hyperparameter optimization.

Parameter	Value
No. of hyperparameters	5
Mean trial to within 1% of final performance	4.42
Median trial to within 1% of final performance	3
Folds still improving in final 25% of trials (%)	8
Mean improvement in final 25% (%)	1.829736152

Figure 6 shows residual diagnostics of the selected CatBoost model based on pooled out-of-fold predictions from 50 outer folds of the Nested CV. The estimates were obtained using pooled estimates derived from the 50 outer folds (5×10) of Nested Cross-

Validation, comprising a total of 61,300 out-of-fold predictions. The prediction–actual value plot showed a generally good agreement between the observed and predicted cooling loads. However, the residual and scale–location plots indicate that the residual variance is not constant across the prediction range and that heteroscedasticity is present. This finding was also confirmed by the Breusch–Pagan test ($LM = 117.80$, $p < 0.001$). The normal Q–Q plot, on the other hand, showed deviations from the theoretical normal distribution, particularly at the tails of the distribution, and this result was consistent with a significant Shapiro–Wilk test ($W = 0.9714$, $p < 0.001$). This result indicates that the assumption of homoscedasticity can be rejected and that the residual variance varies depending on the level of the predicted values. Therefore, it should be noted that estimation errors may vary, particularly in the extreme regions of the cooling load distribution. Consequently, although the selected CatBoost model demonstrates stable performance in out-of-fold predictions, the residuals do not fully satisfy the classical assumptions of homoscedasticity and normality. The model predicts well, but the variance of the prediction errors is not constant, and the residuals are not normally distributed.

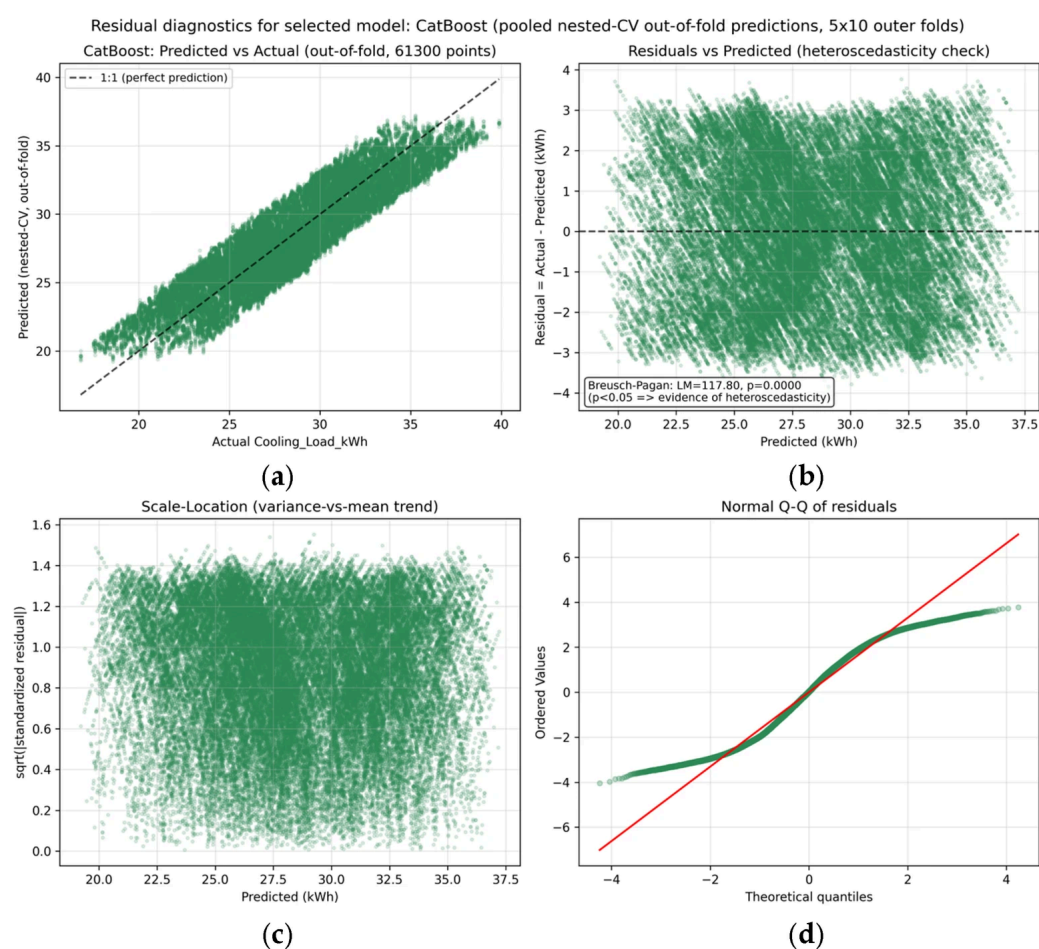


Figure 6. Residual diagnostics of the selected CatBoost model based on pooled out-of-fold predictions from 50 outer folds of the Nested CV. (a) predicted vs. actual, (b) residuals vs. predicted, (c) scale–location, and (d) normal Q–Q plot.

Figure 7 presents the R^2 , RMSE, MAE, and MAPE values obtained from the outer folds of nested cross-validation. Green points represent individual outer-fold values, black points represent the means, and error bars represent the 95% percentile-bootstrap confidence intervals for the mean. The narrow intervals indicate limited variation in the mean outer-fold scores under the repeated partitioning scheme.

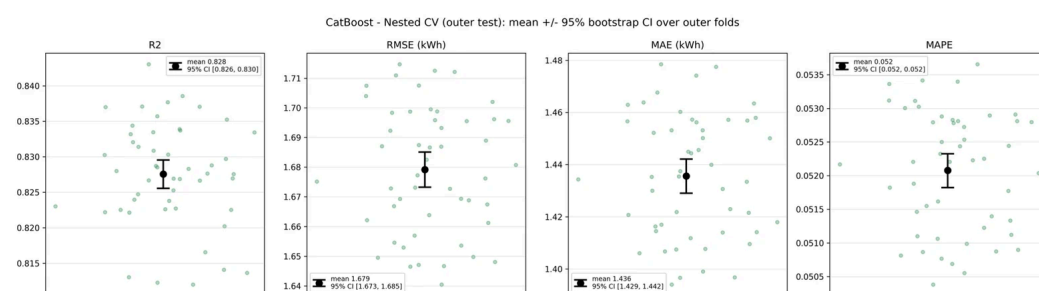


Figure 7. Generalization performance of the CatBoost from Nested Cross-Validation (Outer Test): mean $\pm$ 95% bootstrap confidence interval.

Figure 8 shows the generalization performance of the CatBoost model across the outer folds of nested CV. The feature importance results show that the model's predictions are primarily influenced by Ambient Temperature and Solar Radiation, while Heat Reflective Treatment is the third most important variable. Figure 8b displays the average performance values obtained from the outer folds of nested cross-validation. With an $R^2 = 0.82$, the model explains approximately 82% of the variance in the target variable. The average error values were obtained as RMSE = 1.67 kWh, MAE = 1.43 kWh, and MAPE = 0.052. These results indicate that the model has good prediction accuracy. Figure 8c shows the distribution of R^2 values obtained across the outer folds. The fact that the median value is close to the center of the box and that the box is narrow indicates that R^2 values show low variability across folds. The absence of outliers suggests that model performance is not significantly affected by the data split. Figure 8d shows the distributions of the RMSE and MAE values in the outer folds. Both error metrics have a narrow distribution. This indicates that the model maintains similar error levels across different training–test splits and that its generalization performance is stable.

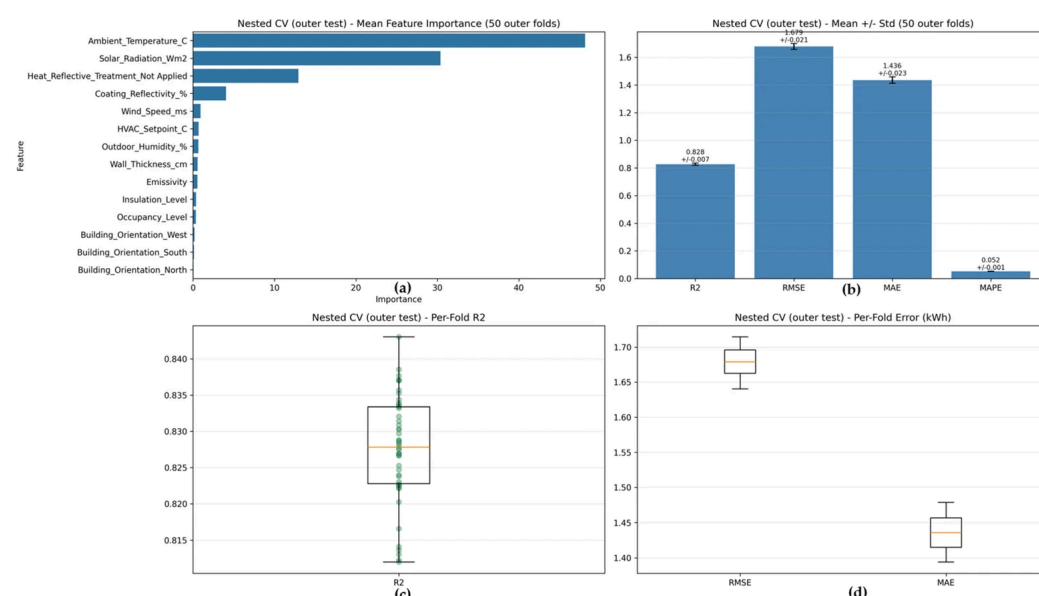


Figure 8. Nested Cross-Validation (Outer Test) performance summary of the CatBoost model. (a) Mean Feature Importance (50 outer folds), (b) Mean $\pm$ Std (50 outer folds), (c) Per fold R^2 , and (d) Per fold Error.

3.2. Model Interpretability

Three complementary approaches were used to evaluate the interpretability of the model: CatBoost's built-in feature importance, SHAP analysis [65], and Nested CV outer-

fold permutation importance. Among these, core feature importances are reported based on the outer-fold permutation importance results, as it assesses outsample generalization. SHAP and built-in feature importance results are presented as supporting explanatory analyses.

To evaluate the variables influencing the predictions of the Selected CatBoost model, permutation importance analysis was performed on test data previously unseen by the model in each outer fold of the Nested CV (Figure 9 and Table 15). This approach allows for the evaluation of the model's outsample generalization behavior by preventing variable importance from stemming solely from fit to the training data. The results showed that Ambient Temperature had the highest permutation importance value (2.4863 ± 0.0273). This was followed by Solar Radiation (1.3055 ± 0.0195) and Heat Reflective Treatment (0.5611 ± 0.0129). Accordingly, while Ambient Temperature was the most dominant variable influencing the model's prediction performance, Solar Radiation and Heat Reflective Treatment also made significant contributions. Permutation importance values were obtained using the CatBoost model retrained on the full data. These results are only illustrative, and the outer-fold permutation importance results were used as the basis for evaluating generalizable feature importance.

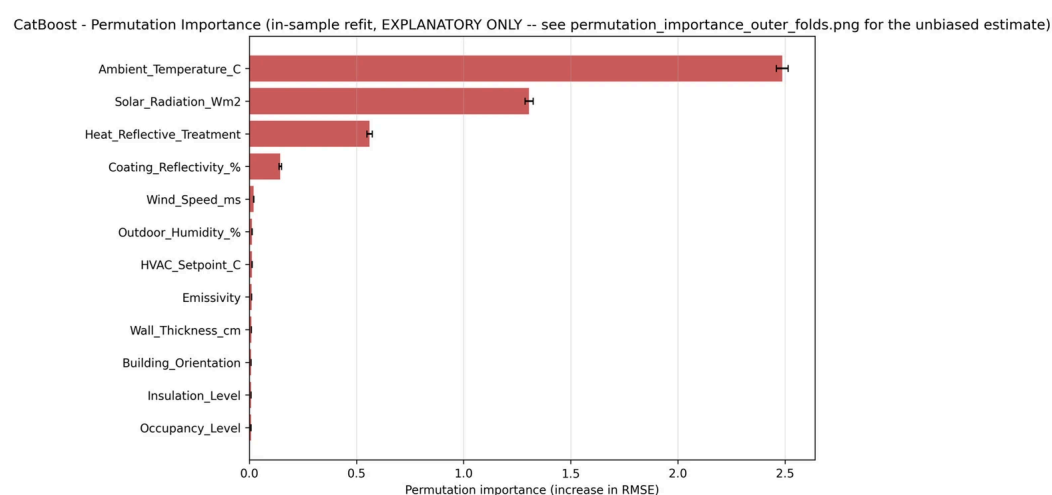


Figure 9. Results of in-sample permutation importance of the CatBoost model (explanatory analysis).

Table 15. CatBoost permutation importance.

Feature	Permutation Importance (Mean)	Permutation Importance (Std)
Ambient Temperature (°C)	2.486300653	0.027367016
Solar Radiation (W/m ²)	1.305545453	0.019561558
Heat Reflective Treatment	0.56118484	0.012912954
Coating Reflectivity (%)	0.144734271	0.005592931
Wind Speed (m/s)	0.020643011	0.000941105
Outdoor Humidity (%)	0.012600869	0.000471581
HVAC Setpoint (°C)	0.012246287	0.000631438
Emissivity	0.011008835	0.000490202
Wall Thickness (cm)	0.009928102	0.000284405
Building Orientation	0.008465251	0.000364779
Insulation Level	0.008103594	0.000393081
Occupancy Level	0.007671578	0.000417263

The permutation importance values of other estimators were found to be close to zero. This indicates that their additional predictive contribution beyond the dominant variables is limited within the current dataset and the established CatBoost model. However, low importance values do not mean that these variables are physically insignificant. Permutations of these variables lead to only a limited change in prediction beyond the information provided by the dominant variables.

The permutation importance results calculated for the nested CV outer folds in the CatBoost model are given in Figure 10. The results are consistent with Table 15.

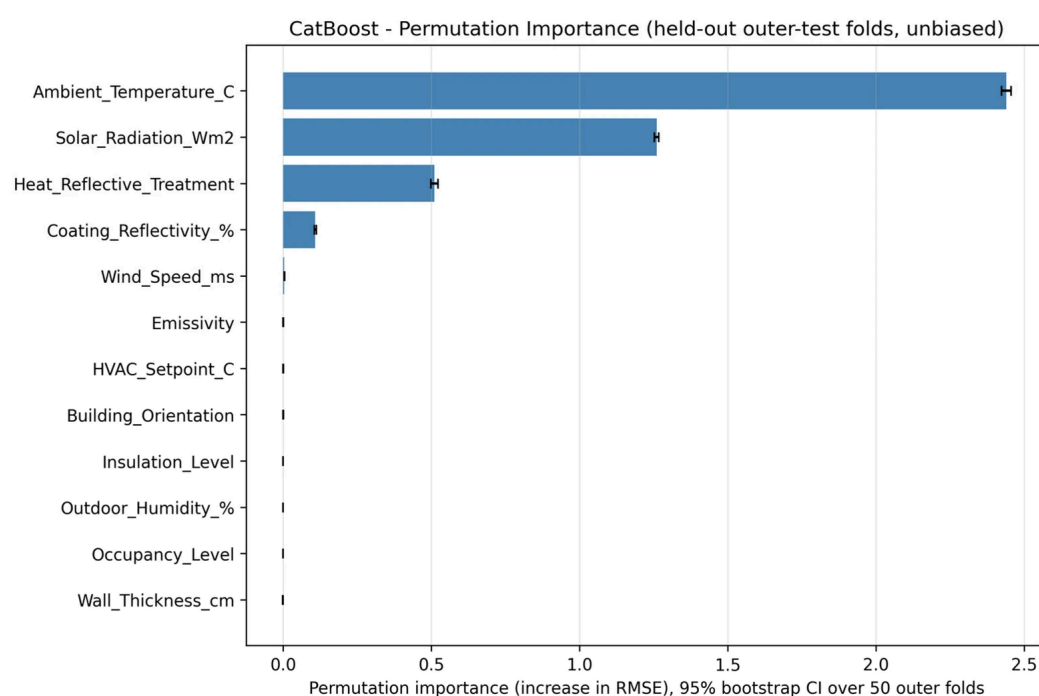


Figure 10. Permutation importance (outer folds).

Table 16 shows the permutation importance results calculated on the outer fold of the CatBoost model. The average permutation importance values represent the contribution of each variable to the model's prediction performance; the standard deviation and 95% bootstrap confidence intervals show the variability of these importance values among the outer folds. This table shows the out-of-fold permutation importance calculated on unseen test data in the outer layers of Nested CVs.

Table 16. Permutation importance (outer folds).

Feature	Permutation Importance (Mean)	Permutation Importance (Std)	CI95 (Low)	CI95 (High)
Ambient Temperature (°C)	2.439963877	0.05716269	2.424052879	2.455459817
Solar Radiation (W/m ²)	1.259989356	0.027659541	1.252450173	1.267575637
Heat Reflective Treatment	0.510309408	0.042045604	0.49888748	0.522210094
Coating Reflectivity (%)	0.108548649	0.013180311	0.104910552	0.112199571
Wind Speed (m/s)	0.00427437	0.002412112	0.003620114	0.004936064
Emissivity	0.00024906	0.001105566	-6.23926×10^{-5}	0.000547285
HVAC Setpoint (°C)	0.000163962	0.001174591	-0.000168702	0.000488404
Building Orientation	8.74555×10^{-5}	0.001054227	-0.000204846	0.000387115
Insulation Level	-0.000258962	0.000944858	-0.000519779	1.78508×10^{-6}

Feature	Permutation Importance (Mean)	Permutation Importance (Std)	CI95 (Low)	CI95 (High)
Outdoor Humidity (%)	−0.000521041	0.001374226	−0.000919293	−0.000150946
Occupancy Level	−0.00056224	0.000990191	−0.000849921	−0.000302278
Wall Thickness (cm)	−0.001038185	0.001547503	−0.001492347	−0.000629696

CatBoost’s built-in feature importance was calculated to evaluate the relative contribution of variables to prediction during the model’s learning process (Table 17). These results are presented as an within-model explanatory analysis; the results of the Nested CV outer-fold permutation importance (Table 16) were used as the basis for evaluating generalizable feature significance.

Table 17. The built-in feature importance of the CatBoost model.

Feature	Importance
Ambient Temperature (°C)	48.1276
Solar Radiation (W/m ²)	30.3842
Heat Reflective Treatment	12.9519
Coating Reflectivity (%)	4.0577
Wind Speed (m/s)	0.9159
HVAC Setpoint (°C)	0.6824
Outdoor Humidity (%)	0.6583
Wall Thickness (cm)	0.5633
Emissivity	0.5351
Insulation Level	0.3875
Occupancy Level	0.3595
Building Orientation (West)	0.1755
Building Orientation (South)	0.1223
Building Orientation (North)	0.0780

In the SHAP analysis (Figure 11), Ambient Temperature, Solar Radiation, and Heat Reflective Treatment also stood out, and this ranking generally agreed with the outer-fold permutation importance results. This consistency shows that the model’s key determinants emerge similarly under different explanatory approaches.

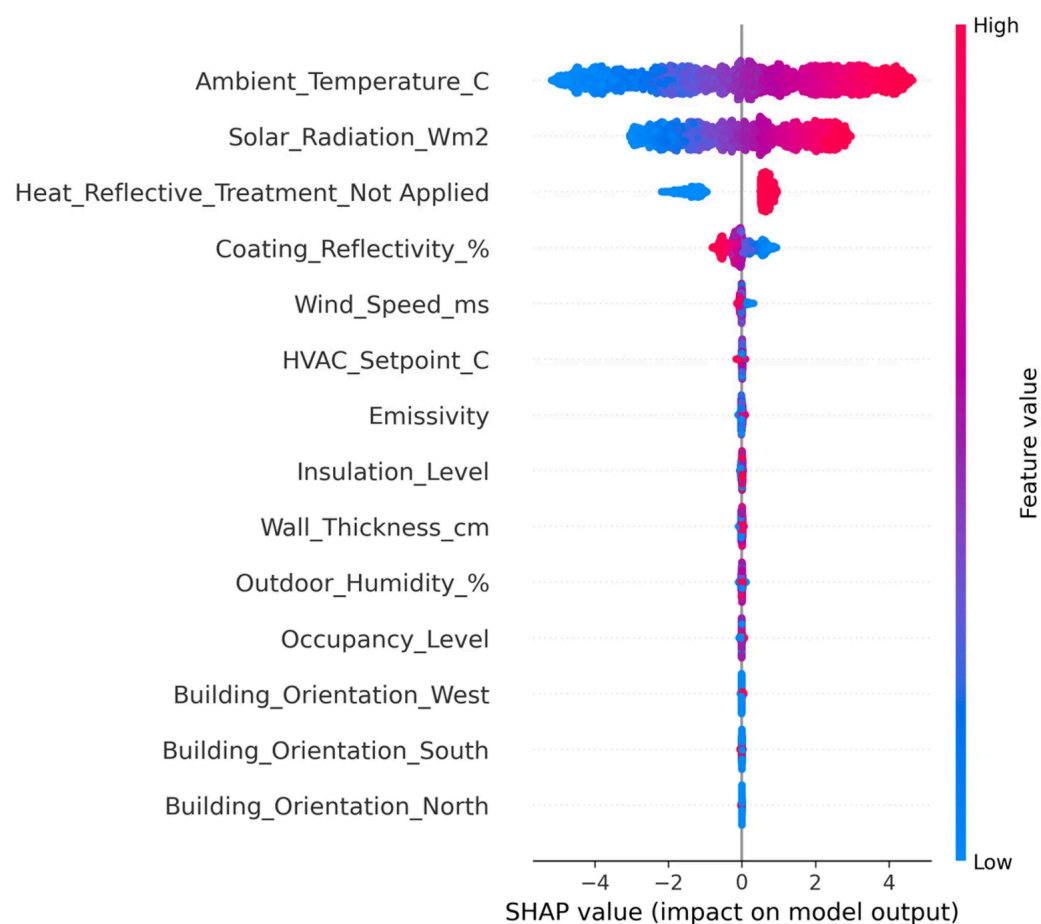


Figure 11. CatBoost SHAP summary.

The MeanAbsSHAP values in Table 18 show the average absolute contribution of each feature to the model predictions. A higher MeanAbsSHAP value indicates that the feature has a greater impact on the model predictions.

Table 18. Mean absolute SHAP values of the features for the selected CatBoost model.

Feature	MeanAbsSHAP
Ambient Temperature (°C)	2.265839755
Solar Radiation (W/m ²)	1.463724105
Heat Reflective Treatment	0.911457972
Coating Reflectivity (%)	0.300489109
Wind Speed (m/s)	0.042909269
HVAC Setpoint (°C)	0.017545331
Emissivity	0.016704828
Insulation Level	0.013589108
Wall Thickness (cm)	0.013140261
Outdoor Humidity (%)	0.012235011
Occupancy Level	0.010543
Building Orientation (West)	0.009452583
Building Orientation (South)	0.006100708
Building Orientation (North)	0.00479071

4. Discussion

In contrast to existing efforts, the process lens of this study extends the study beyond a static comparison of machine-learning models by presenting it as a standardized, repeatable and auditable workflow. Explicitly defining the sequence of preprocessing, tuning, evaluation, comparison and refitting supports reproducibility because each model is processed under the same conditions and the complete analysis can be repeated by other researchers. Process thinking also promotes leanness by revealing redundant activities, unnecessary data handling and duplicated evaluations, allowing the workflow to focus on steps that directly contribute to model selection and interpretation. Moreover, clearly defined task boundaries, decision points and data-partition controls reduce the risk of procedural errors, inconsistent model treatment. Thus, the process perspective strengthens reproducibility, computational efficiency, methodological consistency and the reliability of the reported results.

In this study, different machine learning algorithms were used to estimate building energy consumption. The performance of the models was evaluated using nested cross-validation, not limited to a single training–test split. Furthermore, bootstrap confidence intervals were used to descriptively characterize the variability of the observed outer-fold performance scores. The nested cross-validation framework reduced optimistic bias associated with hyperparameter optimization and provided an estimate of model generalization performance within the evaluated simulation-generated data distribution.

The results showed that nonlinear modeling capabilities could provide an additional prediction advantage for the current dataset. CatBoost outperformed the linear models evaluated in the revised analysis, achieving the lowest Nested-CV RMSE value of 1.6792 ± 0.0210 kWh. However, this finding alone does not prove the existence of specific nonlinear interactions. It suggests that allowing the modeling of nonlinear relationships could improve prediction performance in cooling load prediction.

Permutation importance analysis revealed that some variables (Ambient Temperature (°C) and Solar Radiation (W/m²)) play a more dominant role than others in model predictions. This analysis shows that it is important not only to provide high prediction accuracy but also to ensure that model decisions are explainable. Thus, the proposed approach offers a decision support tool that can be used in practice in terms of both reliable prediction and interpretability.

We acknowledge that using an identical optimization budget across algorithms does not fully account for differences in hyperparameter search-space complexity. More complex models such as XGBoost, LightGBM, and MLP may benefit from larger optimization budgets. In fact, an identical optimization budget was adopted to ensure that each algorithm was allocated the same computational resources. While more complex models possess larger hyperparameter spaces and may potentially benefit from additional trials, the fixed-budget strategy provides a controlled comparison of algorithm performance under equal optimization effort.

One of the strengths of this study is the use of a 5×10 replicate nested cross-validation approach in model evaluation. Unlike the single training–test split commonly used in the literature, nested cross-validation completely separates hyperparameter optimization from performance evaluation. This reduces bias in performance prediction and measures the generalization ability of the model more reliably. In addition, reporting performance based on the distribution obtained from 50 outer test folds instead of just a single value provides a significant advantage that increases the statistical reliability of the results. Model performance was evaluated using R^2 , RMSE, MAE, and MAPE measures together; model selection was made according to the RMSE criterion, which best represents the magnitude of the error. This approach allows for an objective and balanced

comparison of models not only in terms of explanatory power but also in terms of true prediction accuracy. Thus, the evaluation process is not dependent on a single performance measure, and the reliability of the results is increased. To enhance the methodological reliability of the study, optimistic performance estimates obtained during the model development process were not reported as final results. Permutation significance analysis was performed using a model retrained on the entire dataset, solely to account for the contribution of the variables to the model. Similarly, internal cross-validation results used in hyperparameter optimization and performance values obtained from a single training–test split were not included in the final evaluation. Instead, model performance was measured based on results from 50 outer test folds of 5×10 replicate nested cross-validation. This approach provides a statistically robust evaluation method that accurately reflects the true generalization success of the model.

4.1. Scope and Limitations of the Study

The structural characteristics of the dataset used in the study are one of the most fundamental factors that directly influence the learning process of machine learning models and their overall classification performance. The dataset used in this study [30] was created to evaluate the effects of heat-reflective surface treatments applied to building facades on energy performance using a digital twin-based thermal performance analysis approach. The study utilized variables such as ambient temperature, solar radiation, outdoor humidity, wind speed, surface temperatures, wall properties, insulation level, building orientation, occupancy rate, cladding properties, energy consumption, cooling load, and indoor comfort index.

A key limitation of this study was that the dataset used is restricted to specific building characteristics, environmental conditions, and energy performance parameters. In real-world building applications, numerous factors (material properties, operating conditions, etc.) can influence energy performance. Therefore, the results obtained may not be directly generalizable to different climate zones, building types, or usage scenarios. In real application scenarios, many additional factors such as user behavior, HVAC operating strategies, local climate variables, differences in building materials, and building usage scenarios can affect the cooling load. Therefore, the performance of the developed models under different climate zones, different building types, or real operating conditions needs to be further validated.

As a post hoc baseline analysis, an ordinary least squares model using only ambient temperature and solar radiation was evaluated using five-fold cross-validation. This simple two-variable model achieved an R^2 of 0.7714, demonstrating that these two environmental variables alone explain a substantial proportion of the predictable variation in cooling load. However, the best-performing model using the full predictor set achieved a nested cross-validation R^2 of approximately 0.82. Thus, although ambient temperature and solar radiation constitute a strong predictive baseline and appear to dominate the predictive structure of the dataset, they do not fully account for the performance obtained using the complete predictor set. The improvement in out-of-sample performance indicates that the remaining predictors and/or relationships captured by the full-feature model provide additional predictive information beyond the two-variable linear baseline. Accordingly, the dataset is better characterized as being strongly driven by ambient temperature and solar radiation, with additional incremental predictive information contained in the remaining predictors, rather than as a purely two-variable linear function plus residual noise.

Estimating permutation importance reported in this study on the overall dataset, part of which is used for model fitting, reflects feature contributions to the overall

dataset rather than generalization performance. Thus, the reported permutation importance should be interpreted as describing the behavior of the final fitted model rather than feature importance for unseen data.

The dataset provider did not provide building ID, simulation group, or similar grouping information regarding whether the observations belonged to the same building design or a common simulation scenario. Therefore, it was not possible to directly assess the potential dependency structure between observations and apply group-based cross-validation. This constitutes a significant limitation of row-based cross-validation in terms of the independent observation assumption. It is recommended that a more rigorous generalizability assessment of the model be performed using group-based cross-validation in the future, should descriptive information at the building or simulation group level become available.

Since the original dataset description did not specify the temporal aggregation level of the cooling load, the RMSE value (1.6792 kWh) should be considered as the prediction error for each sample in the dataset, not as a direct hourly or annual energy error. Future studies should utilize datasets with clearly defined temporal resolution to allow for more detailed comparisons with building energy simulation applications.

In conclusion, the pipeline machine learning approach proposed in this study provides a reliable ML model evaluation process supported by nested cross-validation, Optuna-based hyperparameter optimization, and multiple performance metrics, ensuring high accuracy in predicting building cooling loads. However, the obtained results were limited by the scope and characteristics of the dataset used.

4.2. Comparison of the Findings with the Literature

A review of the literature on building cooling load estimation reveals that models are generally examined directly. Many studies evaluate model performance using a single training–test split or standard cross-validation, meaning the reported results may not fully reflect the model’s true generalization ability. In this study, different machine learning algorithms were compared under the same evaluation protocol, and the most successful model was determined based on nested cross-validation results.

The models evaluated in this study demonstrated consistent model performance under different data splits. Performing hyperparameter optimization only on internal folds and evaluating performance solely based on results from outer folds ensure an unbiased determination of the model’s true generalization performance. In this respect, the study differs from many studies in the literature by aiming not only to achieve high accuracy but also to provide a statistically reliable and reproducible model evaluation process.

Because the studies used different datasets, target variable definitions, units of measurement, building types, and validation protocols, it is not appropriate to make direct numerical comparisons of the reported performance results. Therefore, rather than ranking the studies in the literature in terms of performance, Table 19 provides a general overview of current approaches in the field of building cooling load and energy prediction. This table presents the chronological development of studies in the literature and aims to show methodological differences in terms of data structures, methods used, and evaluation strategies, rather than directly comparing performance values reported on different datasets.

Table 19. Literature comparison.

Study	Aim	Method	Performance Metric Reported in Original Study
Fan and Ding (2019) [18]	Predict the short-term cooling load in public buildings	Multiple nonlinear regression (MNR)	$R^2 = 0.958$, RMSE = 405.7 kW
Sajjad et al. (2020) [12]	Simultaneously predict heating and cooling loads	Gated recurrent unit (GRU)-based framework	MAE = 0.0167, MSE = 0.0006, RMSE = 0.0247, MAPE = 0.0368
Li et al. (2021) [19]	24-h-ahead building cooling load prediction	Recurrent Neural Network (RNN) architecture with an attention mechanism	RMSE = 524, MAE = 377
Chaganti et al. (2022) [20]	Predict cooling and heating loads	Ensemble model	$R^2 = 0.997$
Kang et al. (2022) [21]	Cooling load prediction	Novel approach	MAE = 389 kW, coefficient of variation (CV) = 12.5%
Lu et al. (2023) [22]	Cooling load prediction of residential building design	AutoML-based framework	$R^2 = 0.9899$ and RMSE = 0.973 kWh/m ²
Alyakoob et al. (2023) [23]	Predict the cooling loads of three university buildings in Arizona	Various machine learning algorithms	Prediction accuracy ¹ score = 98%
Guo et al. (2023) [24]	Predict cooling and heating loads	LightGBM	$R^2 = 0.9924$
Mehdizadeh Khorrami et al. (2024) [25]	Predict the heating and cooling loads	Linear regression, decision trees, logistic regression, and neural networks	Accuracy ¹ = 93.24%
Gao et al. (2024) [26]	Predicting heating and cooling loads	SVR, XGBoost, and a Dempster–Shafer-based ensemble model	$R^2 = 99.36\%$, RMSE = 0.7723
Aydın et al. (2025) [27]	Predict the total heating and cooling loads of buildings with UHP-FRC facade panels	Decision Tree Regression (DTR), Linear Regression (LR), Ridge Regression, K-Nearest Neighbors (KNN), Elastic Net Regression (ENT), Least Absolute Shrinkage and Selection Operator (Lasso) Regression, Histogram Gradient Boosting Regression (HGBR), Support Vector Regression (SVR), eXtreme Gradient Boosting (XGB), Gradient Boosting Regression (GBR), Adaptive Boosting (AB), RF Regression (RFR), Multilayer Perceptron (MLP), Bootstrap Aggregation (Bagging), Voting and Stacking	$R^2 > 0.99$, MAE = 25.46, RMSE = 53.87, NMAE = 0.0012, NRMSE = 0.0027
Peng (2025) [28]	Predicting cooling loads in residential buildings	RF	$R^2 = 0.997$ and RMSE = 0.498
Zhang et al. (2026) [29]	Cooling load prediction	Gray Wolf Optimization (GWO)-CNN-LSTM	$R^2 = 0.9266$, MAE = 218.78 W, RMSE = 327.40 W
This study	Cooling load prediction	18 ML model + Optuna + 5 × 10 Nested CV + Pipeline	$R^2 = 0.8275 \pm 0.0072$, RMSE = 1.6792 ± 0.0210 kWh, MAE = 1.4356 ± 0.0233 kWh, MAPE = 0.0521 ± 0.0009

¹ The term “accuracy” is retained as reported by the original authors. It refers to the prediction performance indicator used in the cited studies and does not imply classification accuracy.

As shown in Table 19, the R^2 values in most studies in the literature fall within the range of 0.95–0.99. In most of these studies, the performance of specific algorithms was reported on a single dataset, or a simple training–test split was used. In this study, however, model selection was performed systematically by using 18 different machine learning algorithms, Optuna-based hyperparameter optimization, a pipeline structure, and a 5×10 nested cross-validation method together. The most successful model yielded a MAPE value of 0.0521 ± 0.0009 . This value indicates the magnitude of the relative prediction error on the evaluated samples. In contrast, the RMSE and MAE values were obtained as 1.6792 ± 0.0210 kWh and 1.4356 ± 0.0233 kWh, respectively, thereby demonstrating the prediction error on both a relative and absolute scale. Instead of focusing solely on achieving the highest accuracy rate, the study presents an approach that prioritizes reliable model evaluation, quantitative reporting of performance uncertainty, and evaluation protocols that provide more robust performance prediction. These results highlight that model reliability should be assessed considering validation methodology, uncertainty quantification, and dataset characteristics rather than only the reported R^2 values.

5. Conclusions

In this study, different machine learning algorithms/models were analyzed using a comprehensive evaluation process based on nested cross-validation. Hyperparameter optimization was performed using Optuna during the training/evaluation process, and model performances were evaluated. The reliability of the results was verified using bootstrap-based 95% confidence intervals.

To ensure model selection was not based on small decimal differences in a single performance value, paired model comparisons, effect sizes, and Holm-corrected tests for multiple comparisons were performed over the same outer folds, and the 1-SE economy rule was considered. As a result of the analyses, CatBoost was selected as the final model. The model obtained $R^2 = 0.8275 \pm 0.0072$, $RMSE = 1.6792 \pm 0.0210$ kWh, $MAE = 1.4356 \pm 0.0233$ kWh, and $MAPE = 0.0521 \pm 0.0009$ over 50 outer folds. Differences between CatBoost and other models were further evaluated using paired outer-fold analyses, thus preventing model selection from being based solely on additional decimal places. The small RMSE differences between CatBoost and strong competitors such as XGBoost, GradientBoosting, and LightGBM indicate that these models have similar prediction performances, but CatBoost stands out as the most suitable model under the revised selection procedure.

In terms of model interpretability, the outer-fold permutation importance results showed that Ambient Temperature was the variable with the highest significance. This was followed by Solar Radiation and Heat Reflective Treatment. However, it was determined that the importance values of other variables were not zero, and therefore their contributions to the dataset could not be completely ignored. SHAP analysis and CatBoost's built-in feature importance results also supported the evaluation of the model's key determinants. These results show that environmental conditions play a dominant role in cooling load prediction, but building and system characteristics can also contribute at lower levels.

Residual identifications showed that the mean of the model residuals was quite close to zero, at -0.0007 kWh. However, the Breusch–Pagan test revealed that the residual variance was dependent on the estimated value, and the Shapiro–Wilk test showed that the residuals deviated from a normal distribution. Therefore, despite the model's high prediction performance, it was concluded that the assumptions of constant variance and residual normality were not fully met, and the prediction uncertainty should be

carefully considered, especially in the extreme regions of the cooling load distribution.

Finally, the computational cost and reproducibility of the study were also clearly evaluated. The experimental protocol, which included a total of 18 models, 50 outer folds, and 50 Optuna trials for each outer fold, took approximately 32,546.5 s (9.04 h). The computational environment, library versions used, random seeds, and experimental settings are also reported to support the reproducibility of the study.

Overall, this study presents a more reliable assessment approach to building cooling load estimation that utilizes Nested Cross-Validation, hyperparameter optimization, paired outer-fold comparisons, 1-SE model selection, feature importance analysis, and residual diagnostics together, rather than simply comparing models to find the lowest error value. The results demonstrate that CatBoost can provide robust prediction performance on the current dataset. However, it should be noted that the results are limited by the characteristics of the simulation-based dataset; in the future, the external validity of the model should be further tested using real building operational data, different climate zones, and different building types.

Future work will test the proposed evaluation approach with different datasets covering different climate zones, building types, and with real-world field data, to assess the generalizability of the model. Studies using larger and more diverse datasets will test the reliability of the pipeline in different application scenarios and increase its applicability in building energy management systems.

Supplementary Materials: The following supporting information can be downloaded at <https://www.mdpi.com/article/doi/s1>, Table S1: Pairwise statistical comparisons of model performance based on paired outer-fold RMSE values. Table S2: Hyperparameter search spaces (Optuna, TPE, 50 trials, inner 5-fold CV).

Author Contributions: Conceptualization, G.B. and Y.A.; methodology, Ü.I.; software, Ü.I.; validation, Ü.I.; investigation, S.M.N.; data curation, Y.A.; writing—original draft preparation, S.M.N., Y.A. and Ü.I.; writing—review and editing, S.M.N., G.B., W.-W.K. and Z.W.G.; visualization, Ü.I. and Y.A.; supervision, Z.W.G. All authors have read and agreed to the published version of the manuscript.

Funding: This research was supported by the Korea Institute of Energy Technology Evaluation and Planning (KETEP) and funded by the Ministry of Trade, Industry and Energy (MOTIE) of the Republic of Korea. (RS-2023-00236325, RS-2024-00441420, RS-2024-00442817).

Data Availability Statement: The data presented in this study are available on request from the corresponding authors.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. International Energy Agency (IEA). *Energy Efficiency 2025*; International Energy Agency: Paris, France, 2025. Available online: <https://www.iea.org/reports/energy-efficiency-2025/buildings> (accessed on 4 June 2026). [Google Scholar]
2. International Energy Agency (IEA). *Global Energy Review 2026*; International Energy Agency: Paris, France, 2026. Available online: <https://www.iea.org/reports/global-energy-review-2026/global-trends> (accessed on 4 June 2026). [Google Scholar]
3. Yuan, F.; Yao, R.; Sadrizadeh, S.; Li, B.; Cao, G.; Zhang, S.; Zhou, S.; Liu, H.; Bogdan, A.; Croitoru, C.; et al. Thermal comfort in hospital buildings—A literature review. *J. Build. Eng.* **2022**, *45*, 103463. [Google Scholar] [CrossRef]
4. Cao, L.; Li, T.; Wang, F. Heat insulation and thermal insulation method of passive low energy consumption residential building exterior envelope structure based on BIM. *Results Eng.* **2024**, *23*, 102734. [Google Scholar] [CrossRef]
5. Arachchi Appuhamilage, P.D.T.; Rijal, H.B.; Shahzad, S. Localized thermal control for global impact: A meta-analytical review of thermal comfort and energy performance of personal comfort systems. *Appl. Energy* **2026**, *403*, 127051. [Google Scholar] [CrossRef]

6. Ihara, T.; Gustavsen, A.; Jelle, B.P. Effect of facade components on energy efficiency in office buildings. *Appl. Energy* **2015**, *158*, 422–432. [Google Scholar] [CrossRef]
7. Aryal, N.; Ottosen, L.D.M.; Kofoed, M.V.W.; Pant, D. *Emerging Technologies and Biological Systems for Biogas Upgrading*, 1st ed.; Academic Press: Amsterdam, The Netherlands, 2021; Volume 530. [Google Scholar] [CrossRef]
8. Kim, S.-C.; Shin, H.-I.; Ahn, J. Energy performance analysis of airport terminal buildings by use of architectural, operational information and benchmark metrics. *J. Air Transp. Manag.* **2020**, *83*, 101762. [Google Scholar] [CrossRef]
9. Calama-González, C.M.; León-Rodríguez, Á.L.; Suárez, R. Daylighting Performance of Solar Control Films for Hospital Buildings in a Mediterranean Climate. *Energies* **2019**, *12*, 489. [Google Scholar] [CrossRef]
10. Yoshino, H.; Hong, T.; Nord, N. IEA EBC annex 53: Total energy use in buildings—Analysis and evaluation methods. *Energy Build.* **2017**, *152*, 124–136. [Google Scholar] [CrossRef]
11. Abdel-Jaber, F.; Dirks, K.N. A Review of Cooling and Heating Loads Predictions of Residential Buildings Using Data-Driven Techniques. *Buildings* **2024**, *14*, 752. [Google Scholar] [CrossRef]
12. Sajjad, M.; Khan, S.U.; Khan, N.; Haq, I.U.; Ullah, A.; Lee, M.Y.; Baik, S.W. Towards Efficient Building Designing: Heating and Cooling Load Prediction via Multi-Output Model. *Sensors* **2020**, *20*, 6419. [Google Scholar] [CrossRef] [PubMed]
13. Hu, M.; Wan, E.; Qiao, Y.; Xu, S.; Li, S.; Zhu, R.; Guo, J.; Chen, X.; Zeng, H. Transient interference-based spectroscopy for a molecular-bond-sensitive probe of lithium ion batteries. *Energy Environ. Sci.* **2026**, *19*, 2889–2899. [Google Scholar] [CrossRef]
14. Jin, H.; Hu, M.; Wan, E.; Qiao, Y.; Duan, J.; Cheng, L.; Zeng, H. Density-Clustering Decoding of Plasma Micro-Region Dynamic Molecular Fingerprint Spectra. *Laser Photon. Rev.* **2026**, *early view*. [Google Scholar] [CrossRef]
15. Cakiroglu, C.; Aydın, Y.; Bekdaş, G.; Isikdag, U.; Sadeghifam, A.N.; Abualigah, L. Cooling load prediction of a double-story terrace house using ensemble learning techniques and genetic programming with SHAP approach. *Energy Build.* **2024**, *313*, 114254. [Google Scholar] [CrossRef]
16. Duro, B.; Gomes, A.; Correia, F.B.; Borges, A.R.; Bernardino, J. Machine Learning and Deep Learning for Dropout Prediction in Higher Education: A Review. *Computers* **2026**, *15*, 164. [Google Scholar] [CrossRef]
17. Alanazi, A. Using machine learning for healthcare challenges and opportunities. *Inform. Med. Unlocked* **2022**, *30*, 100924. [Google Scholar] [CrossRef]
18. Fan, C.; Ding, Y. Cooling load prediction and optimal operation of HVAC systems using a multiple nonlinear regression model. *Energy Build.* **2019**, *197*, 7–17. [Google Scholar] [CrossRef]
19. Li, A.; Xiao, F.; Zhang, C.; Fan, C. Attention-based interpretable neural network for building cooling load prediction. *Appl. Energy* **2021**, *299*, 117238. [Google Scholar] [CrossRef]
20. Chaganti, R.; Rustam, F.; Daghriri, T.; Díez, I.d.I.T.; Mazón, J.L.V.; Rodríguez, C.L.; Ashraf, I. Building Heating and Cooling Load Prediction Using Ensemble Machine Learning Model. *Sensors* **2022**, *22*, 7692. [Google Scholar] [CrossRef] [PubMed]
21. Kang, X.; Wang, X.; An, J.; Yan, D. A novel approach of day-ahead cooling load prediction and optimal control for ice-based thermal energy storage (TES) system in commercial buildings. *Energy Build.* **2022**, *275*, 112478. [Google Scholar] [CrossRef]
22. Lu, C.; Li, S.; Reddy Penaka, S.; Olofsson, T. Automated machine learning-based framework of heating and cooling load prediction for quick residential building design. *Energy* **2023**, *274*, 127334. [Google Scholar] [CrossRef]
23. Alyakoob, A.; Hartono, S.; Johnson, T.; Middel, A. Estimating cooling loads of Arizona State University buildings using micro-climate data and machine learning. *J. Build. Eng.* **2023**, *64*, 105705. [Google Scholar] [CrossRef]
24. Guo, J.; Yun, S.; Meng, Y.; He, N.; Ye, D.; Zhao, Z.; Jia, L.; Yang, L. Prediction of heating and cooling loads based on light gradient boosting machine algorithms. *Build. Environ.* **2023**, *236*, 110252. [Google Scholar] [CrossRef]
25. Mehdizadeh Khorrami, B.; Soleimani, A.; Pinnarelli, A.; Brusco, G.; Vizza, P. Forecasting heating and cooling loads in residential buildings using machine learning: A comparative study of techniques and influential indicators. *Asian J. Civ. Eng.* **2024**, *25*, 1163–1177. [Google Scholar] [CrossRef]
26. Gao, T.; Han, X.; Wang, J.; Geng, Y.; Zhang, H.; Song, T. Enhancing building energy efficiency: An integrated approach to predicting heating and cooling loads using machine learning and optimization algorithms. *J. Build. Eng.* **2024**, *98*, 110759. [Google Scholar] [CrossRef]
27. Aydın, Y.; Cakiroglu, C.; Bekdaş, G.; Abualigah, L.; Geem, Z.W. Predicting the total heating and cooling energy consumption with ultra high-performance fiber reinforced concrete facade panels using explainable machine learning models. *Energy Rep.* **2025**, *14*, 3254–3275. [Google Scholar] [CrossRef]
28. Peng, P. Predicting residential building cooling load with a machine learning random forest approach. *Int. J. Interact. Des. Manuf. (IJIDeM)* **2025**, *19*, 3421–3434. [Google Scholar] [CrossRef]

29. Zhang, X.; Zhang, C.; Ma, Y.; Liu, K. Building Cooling Load Prediction Based on GWO-CNN-LSTM. *Energies* **2026**, *19*, 498. [Google Scholar] [CrossRef]
30. Colabss. Heat Reflective Building Energy Dataset. Kaggle. Available online: <https://www.kaggle.com/datasets/colabsss/heat-r-reflective-building-energy-dataset> (accessed on 15 June 2026).
31. Heat Reflective Building Energy Dataset for Cooling Load Prediction. Available online: <https://zenodo.org/records/21894835> (accessed on 11 August 2026).
32. Goodfellow, I.; Bengio, Y.A. *Courville, Deep Learning*; The MIT Press: Cambridge, UK, 2016. [Google Scholar]
33. Scikit-Learn Developers. OneHotEncoder. Scikit-Learn. Available online: <https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.OneHotEncoder.html> (accessed on 20 June 2026).
34. Benyasrisawat, P.; Ounlert, N. Fraud Risk and Audit Opinions Across Countries: Complementing Accounting-Based Fraud Risk with Machine Learning Methods. *J. Risk Financ. Manag.* **2026**, *19*, 26. [Google Scholar] [CrossRef]
35. Huang, S. Linear regression analysis. In *International Encyclopedia of Education*, 4th ed.; Elsevier: Amsterdam, The Netherlands, 2023; pp. 548–557. [Google Scholar] [CrossRef]
36. Tibshirani, R. Regression Shrinkage and Selection Via the Lasso. *J. R. Stat. Soc. Ser. B Stat. Methodol.* **1996**, *58*, 267–288. [Google Scholar] [CrossRef]
37. Akhtar, N.; Alharthi, M.F.; Khan, M.S. Mitigating Multicollinearity in Regression: A Study on Improved Ridge Estimators. *Mathematics* **2024**, *12*, 3027. [Google Scholar] [CrossRef]
38. Zou, H.; Hastie, T. Regularization and Variable Selection Via the Elastic Net. *J. R. Stat. Soc. Ser. B Stat. Methodol.* **2005**, *67*, 301–320. [Google Scholar] [CrossRef]
39. Huber, P.J. *Robust Estimation of a Location Parameter*; Springer: New York, NY, USA, 1992; pp. 492–518. [Google Scholar] [CrossRef]
40. Halder, R.K.; Uddin, M.N.; Uddin, M.A.; Aryal, S.; Khraisat, A. Enhancing K-nearest neighbor algorithm: A comprehensive review and performance analysis of modifications. *J. Big Data* **2024**, *11*, 113. [Google Scholar] [CrossRef]
41. Roy, K.; Kar, S.; Das, R.N. Selected Statistical Methods in QSAR. In *Understanding the Basics of QSAR for Applications in Pharmaceutical Sciences and Risk Assessment*; Elsevier: Amsterdam, The Netherlands, 2015; pp. 191–229. [Google Scholar] [CrossRef]
42. Montesinos López, O.A.; Montesinos López, A.; Crossa, J. Fundamentals of Artificial Neural Networks and Deep Learning. In *Multivariate Statistical Machine Learning Methods for Genomic Prediction*; Springer International Publishing: Cham, Switzerland, 2022; pp. 379–425. [Google Scholar] [CrossRef]
43. Prokhorenkova, L.; Gusev, G.; Vorobev, A.; Dorogush, A.V.; Gulin, A. CatBoost: Unbiased boosting with categorical features. *arXiv* **2017**, arXiv:1706.09516. [Google Scholar]
44. Ahmed Soomro, A.; Akmar Mokhtar, A.; Hussin, H.B.; Lashari, N.; Lekan Oladosu, T.; Muslim Jameel, S.; Inayat, M. Analysis of machine learning models and data sources to forecast burst pressure of petroleum corroded pipelines: A comprehensive review. *Eng. Fail. Anal.* **2024**, *155*, 107747. [Google Scholar] [CrossRef]
45. Pan, Z.; Lu, W.; Bai, Y. Groundwater contaminated source estimation based on adaptive correction iterative ensemble smoother with an auto lightgbm surrogate. *J. Hydrol.* **2023**, *620*, 129502. [Google Scholar] [CrossRef]
46. Chen, T.; Guestrin, C. XGBoost. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*; ACM: New York, NY, USA, 2016; pp. 785–794. [Google Scholar] [CrossRef]
47. Gamil, S.; Zeng, F.; Alrifay, M.; Asim, M.; Ahmad, N. An Efficient AdaBoost Algorithm for Enhancing Skin Cancer Detection and Classification. *Algorithms* **2024**, *17*, 353. [Google Scholar] [CrossRef]
48. Belyadi, H.; Haghighat, A. Supervised Learning. In *Machine Learning Guide for Oil and Gas Using Python*; Elsevier: Amsterdam, The Netherlands, 2021; pp. 169–295. [Google Scholar] [CrossRef]
49. Theerthagiri, P. Liver disease classification using histogram-based gradient boosting classification tree with feature selection algorithm. *Biomed. Signal Process. Control* **2025**, *100*, 107102. [Google Scholar] [CrossRef]
50. Yu, Z.; Haghighat, F.; Fung, B.C.M.; Yoshino, H. A decision tree method for building energy demand modeling. *Energy Build.* **2010**, *42*, 1637–1646. [Google Scholar] [CrossRef]
51. Breiman, L. Random Forests. *Mach. Learn.* **2001**, *45*, 5–32. [Google Scholar] [CrossRef]
52. Magalhães, B.; Bento, P.; Pombo, J.; Calado, M.d.R.; Mariano, S. Short-Term Load Forecasting Based on Optimized Random Forest and Optimal Feature Selection. *Energies* **2024**, *17*, 1926. [Google Scholar] [CrossRef]

53. Aziz, T.; Camana, M.R.; Garcia, C.E.; Hwang, T.; Koo, I. REM-Based Indoor Localization with an Extra-Trees Regressor. *Electronics* **2023**, *12*, 4350. [Google Scholar] [CrossRef]
54. Lee, T.-H.; Ullah, A.; Wang, R. *Bootstrap Aggregating and Random Forest*; Springer International Publishing: Cham, Switzerland, 2020; pp. 389–429. [Google Scholar] [CrossRef]
55. Gupta, A.; Kulkarni, M.; Mukherjee, A. Accurate prediction of B-form/A-form DNA conformation propensity from primary sequence: A machine learning and free energy handshake. *Patterns* **2021**, *2*, 100329. [Google Scholar] [CrossRef] [PubMed]
56. Akiba, T.; Sano, S.; Yanase, T.; Ohta, T.; Koyama, M. Optuna. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*; ACM: New York, NY, USA, 2019; pp. 2623–2631. [Google Scholar] [CrossRef]
57. Flores, B.E. A pragmatic view of accuracy measurement in forecasting. *Omega* **1986**, *14*, 93–98. [Google Scholar] [CrossRef]
58. Nair, S.M.; Nammalvar, A.; Andrushia, D. Data-Driven Prediction of Compressive Strength in Concrete with Lightweight Expanded Clay Aggregate Using Machine Learning Techniques. *J. Compos. Sci.* **2026**, *10*, 151. [Google Scholar] [CrossRef]
59. Harris, C.R.; Millman, K.J.; van der Walt, S.J.; Gommers, R.; Virtanen, P.; Cournapeau, D.; Wieser, E.; Taylor, J.; Berg, S.; Smith, N.J.; et al. Array programming with NumPy. *Nature* **2020**, *585*, 357–362. [Google Scholar] [CrossRef] [PubMed]
60. McKinney, W. Data Structures for Statistical Computing in Python. In *Proceedings of the 9th Python in Science Conference*; Walt, S., Millman, J., Eds.; SciPy: Austin, TX, USA, 2010; pp. 56–61. [Google Scholar] [CrossRef]
61. Pedregosa, F.; Varoquaux, G.; Gramfort, A.; Michel, V.; Thirion, B.; Grisel, O.; Blondel, M.; Prettenhofer, P.; Weiss, R.; Dubourg, V.; et al. Scikit-learn: Machine learning in Python. *J. Mach. Learn. Res.* **2011**, *12*, 2825–2830. [Google Scholar]
62. Ke, G.; Meng, Q.; Finley, T.; Wang, T.; Chen, W.; Ma, W.; Ye, Q.; Liu, T. LightGBM: A highly efficient gradient boosting decision tree. In *Proceedings of the 31st International Conference on Neural Information Processing Systems (NIPS'17)*; Curran Associates Inc.: Red Hook, NY, USA, 2017; pp. 3149–3157. [Google Scholar]
63. Virtanen, P.; Gommers, R.; Oliphant, T.E.; Haberland, M.; Reddy, T.; Cournapeau, D.; Burovski, E.; Peterson, P.; Weckesser, W.; Bright, J.; et al. SciPy 1.0: Fundamental algorithms for scientific computing in Python. *Nat. Methods* **2020**, *17*, 261–272. [Google Scholar] [CrossRef] [PubMed]
64. Uisikdag. Robust_Cooling_Load. Github. Available online: https://github.com/uisikdag/robust_cooling_load (accessed on 12 August 2026).
65. Lundberg, S.M.; Lee, S.I. A unified approach to interpreting model predictions. In *Proceedings of the Advances in Neural Information Processing Systems*; Curran Associates, Inc.: Long Beach, CA, USA, 2017. [Google Scholar]

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.