Research Square

Preprints are preliminary reports that have not undergone peer review.
They should not be considered conclusive, used to inform clinical practice,
or referenced by the media as validated information.

Predicting Thermal Energy Demands for Residential Buildings in Egypt: A Machine Learning Perspective

Shereen El-Shekheby
Port Said University
Basma Nashaat
basma_nashaat@eng.psu.edu.eg
Port Said University
Mona Nashaat
Port Said University

Research Article

Keywords: Residential buildings, Heating loads, Cooling loads, Machine learning, Supervised learning

Posted Date: November 12th, 2026

DOI: https://doi.org/10.21203/rs.3.rs-11411710/v1

License: This work is licensed under a Creative Commons Attribution 4.0 International License.
Read Full License

Additional Declarations: No competing interests reported.

Predicting Thermal Energy Demands for Residential Buildings in Egypt: A Machine Learning Perspective

Abstract

Buildings account for a significant share of global energy use, with heating and cooling loads representing the most significant contributors to residential energy demand. Accurate prediction of these loads is therefore essential for designing efficient HVAC systems and supporting energy-saving strategies. Traditional estimation methods, however, rely on mathematical models or empirical correlations that often fail to capture the complex interactions among building design variables, occupancy patterns, and environmental conditions. This study addresses this gap by conducting a comprehensive experimental assessment of multiple ML algorithms for predicting residential heating and cooling loads. A simulated dataset was generated from diverse building configurations, assuming a residential setting located in Cairo, Egypt, varying in geometry, orientation, insulation, and glazing parameters. Data preprocessing, normalization, and systematic feature selection were applied to identify the most relevant predictors. Nine supervised learning methods, including ensemble, kernel-based, and linear models, were trained and evaluated using rigorous cross-validation and multiple performance metrics. The findings indicate that ensemble-based approaches offer superior predictive performance, while feature ranking highlights the strong influence of window-to-wall ratio and glazing distribution. These results emphasize the role of data-driven methods in enhancing energy-efficient building design and provide actionable insights for practitioners in architecture, engineering, and energy management.

Keywords

Residential buildings; Heating loads; Cooling loads; Machine learning; Supervised learning

1. Introduction

The growing demand for residential energy is primarily influenced by population growth, urbanization, and the increasing use of energy-intensive appliances and devices [1], [2]. Studies show that buildings are responsible for nearly 40% of global energy consumption [3], with heating, ventilation, and air conditioning (HVAC) systems contributing substantially due to their vital role in ensuring indoor comfort. This rising energy demand places heavy pressure on power infrastructure and raises environmental concerns, drawing the attention of policymakers, building occupants, and design professionals [4].

The heating and cooling requirements of buildings are a major contributor to their overall energy consumption. Heating and cooling loads refer to the energy needed to maintain indoor temperatures, either by adding warmth during colder periods or reducing heat during warmer months [5]. These loads are influenced by several factors, including a building's design, the thermal properties of construction materials, the effectiveness of insulation, and external weather conditions. Since maintaining thermal comfort can require substantial energy, careful management of these factors is essential [6]. Thoughtful design strategies, advanced insulation techniques, and the use of modern HVAC systems can help reduce energy demand, lower utility costs, and lessen environmental impact. By addressing these aspects, buildings can achieve more efficient energy use and support broader sustainability goals.

Estimating a building's heating and cooling loads is essential for designing efficient HVAC systems and insulation strategies. Accurate assessments can reduce energy waste by avoiding oversized equipment and ensuring efficient energy distribution [7]. It also identifies opportunities to lower energy consumption, such as improving insulation or fine-tuning system settings [8]. As a result, the building operates more efficiently, reducing overall energy use and enhancing performance. Accurate prediction of energy loads, covering both heating and cooling demands, is a critical step toward achieving long-term energy efficiency and optimizing building functionality. A key method for achieving high-efficiency buildings is performance-driven design, which estimates the energy required to maintain occupant comfort using physics-based simulation tools [9]. These simulations allow architects and engineers to evaluate how a building's design, materials, and systems affect its thermal performance before construction begins. Traditionally, reaching an optimal design through simulation involves an iterative cycle of design, analysis, and revision [10]. Although effective, this process can be time-consuming and may restrict the range of explored design alternatives. Integrating optimization techniques into performance-driven design broadens the set of possible solutions, enabling a more comprehensive exploration of energy-efficient strategies during the early design phase.

Traditional approaches [11] rely heavily on mathematical equations and empirical correlations. While useful, they often fail to reflect the complex interdependencies among the diverse factors that influence energy consumption .These factors include building design characteristics, such as insulation type and thickness, window orientation and glazing, and overall architectural layout [12]. Occupancy patterns, including variations in the number of occupants and their behavior, also significantly affect energy use. Weather conditions—such as temperature, humidity, and solar radiation—further impact heating and cooling demands. Additionally, operational schedules, covering the timing and intensity of lighting, HVAC operation, and equipment use, add further complexity to energy consumption predictions [13]. Consequently, the accuracy of traditional estimation methods can be limited, potentially leading to inefficient energy management and higher operational costs [14].

In recent years, machine learning (ML) techniques have gained attention for their ability to address complex design challenges, including building planning and energy performance prediction [1]. For example, Kumar et al. [15] examine the use of several neural networks to predict the energy performance of residential buildings. The study evaluates different data mining algorithms, comparing them against a proposed deep learning method [15] to assess their predictive accuracy. Another study [16] compares physics-informed Autoregressive-Moving-Average with Exogenous Inputs (ARMAX) models to ML models based on Random Forests and Input Convex Neural Networks through experiments on a practical building application. The work seeks to minimize energy consumption while maintaining occupant comfort, demonstrating that predictive control can achieve considerable energy savings while maintaining satisfactory performance across all models.

While these methods show significant potential, they face challenges related to the availability and quality of training data. As supervised, data-driven approaches, ML algorithms rely on diverse and comprehensive datasets to develop and validate accurate models. However, existing datasets often fail to capture the full range of variables influencing building energy consumption. Most studies [15], [16], [17] do not provide specific details about the datasets used, making it difficult to evaluate the generalizability of their findings. Moreover, many investigations [18] do not incorporate domain knowledge into model training, instead favoring models with lower computational demands and better sample efficiency. These limitations underscore the need for

further research aimed at developing advanced tools, methods, and policies to support architects and engineers in designing energy-efficient buildings. Bridging this data gap is crucial for realizing the full potential of ML and advancing the development of more sustainable and comfortable built environments.

Therefore, this study presents a detailed experimental evaluation of various machine learning techniques for estimating residential buildings' cooling and heating loads. This research highlights the critical influence of dataset selection on model performance. Our analysis incorporates a wide range of building parameters—such as insulation levels, window types, orientation, building materials, and occupancy patterns—. By systematically assessing the influence of these variables, this research demonstrates how dataset diversity directly impacts model reliability. The findings emphasize the necessity of using datasets that reflect the diverse and complex factors influencing building energy consumption. This approach enhances model robustness and offers practical guidance for architects, engineers, and energy managers aiming to improve energy efficiency in residential buildings.

This paper is structured as follows: Section 2 reviews the related literature on building energy prediction, highlighting the strengths and limitations of existing approaches. Section 3 describes the dataset construction, preprocessing, and feature engineering steps applied before model development. Section 4 presents the machine learning models selected for analysis, outlining their main characteristics and training procedures. Section 5 details the evaluation methodology and the experimental results, comparing model performances and analyzing feature importance. Finally, Section 6 concludes the study and outlines directions for future research.

2. Related Work

Recent studies have explored the use of machine learning models to predict heating and cooling loads in residential buildings, aiming to support improvements in energy efficiency. This section reviews existing work, grouped by methodological focus, and highlights the remaining challenges that motivate the present research. Table 1 provides a summary of the datasets, building types, locations, sample sizes, models, and prediction targets reviewed in this section.

A widely used dataset [19] contains 768 samples with eight input features—relative compactness, surface area, wall area, roof area, overall height, orientation, glazing area, and glazing area distribution—and two outputs: heating load (HL) and cooling load (CL). Early studies applied classical ML techniques to this dataset. For instance, one study [20] compared linear regression and random forest (RF) models, using RF to evaluate feature importance. Interestingly, although the glazing area was identified as the most influential feature, it did not exhibit the strongest association with HL or CL.

Building on these findings, another research [21] proposed an extreme gradient boosting (XGBoost) model, achieving high accuracy with R^2 scores of 0.9993 for HL and 0.998 for CL, evaluated using standard error metrics such as RMSE, MAE, and MAPE. Similarly, Mehdizadeh Khorrami et al. [22] employed linear regression and decision tree classification methods, using Pearson correlation coefficients to assess input feature impacts. Decision trees achieved the highest accuracies, with 98.96% for heating load predictions and 93.24% for cooling load predictions.

Mohebbi and Afzal [23] further explored regression techniques by applying support vector regression, decision tree regression, and extra tree regression. They introduced an optimization strategy, the Tyrannosaurus Rex algorithm, which enhanced model performance to an R^2 of 0.987 and an RMSE of 1.167. However, these studies primarily focused on model accuracy without addressing generalization across diverse building configurations or climates.

To improve predictive performance and robustness, hybrid and ensemble methods have been explored. Sadaghat et al. [24] developed the XACM model, combining Extreme Gradient Boosting (XGBoost), Adaptive Boosting (AdaBoost), and Covariance Matrix Adaptation Evolution Strategy (CMAES). Their approach accounted for various indoor environmental conditions and building parameters in Athens. While demonstrating strong predictive capability, the model's dependency on specific assumptions, such as standardized façade properties and occupancy patterns, may restrict its applicability to other environments.

Other work [25] combined a Multi-Layer Perceptron (MLP) with meta-heuristic optimization techniques, namely Particle Swarm Optimization (PSO) and Grey Wolf Optimizer (GWO). Although the MLP-PSOGWO model showed robust performance, its computational demands during training and optimization present challenges for practical deployment.

In parallel, some studies have focused on predicting broader building energy consumption patterns rather than specifically targeting heating and cooling loads. Dong et al. [26] developed an ensemble model combining support vector regression (SVR), artificial neural networks (ANN), and multiple linear regression (MLR), using data from 507 nonresidential buildings in New York [27]. Their ensemble model outperformed individual models but did not account for variations across building types or climatic conditions.

Similarly, Olu-Ajayi et al. [28] compared nine ML techniques, including deep neural networks (DNN), K-nearest neighbors (KNN), and RF, using a dataset of 5,000 residential buildings in the United Kingdom. DNN models achieved the best performance; however, the study focused on annual energy use without a detailed analysis of heating or cooling demands.

Several studies have emphasized feature selection and causal analysis. Dinmohammadi et al. [29] proposed a PSO-optimized random forest model to identify variables influencing heating energy consumption. Using causal inference techniques, they demonstrated links between air temperature, water pipe temperature, and energy use. Despite providing valuable insights, the study lacked extensive comparative evaluation.

In addition, research on online learning methods [30] introduced extreme learning machines (ELM) and online sequential ELM (OSELM) for real-time prediction of HL and CL. A total of 24 models were developed, and while the ELM models achieved low MAE values, the experiments primarily assessed computational efficiency rather than predictive generalization.

Wang et al. [31] proposed a stacking model using input features from two educational buildings in Tianjin, China. The model categorized input variables into meteorological, time-related, and historical consumption data. However, the reliance on a small sample raises concerns about generalizability to broader building types and climatic regions.

Although considerable progress has been made in predicting building energy loads using machine learning [20], [20], [21], [22], [23] , several limitations remain. Many studies rely on a single dataset [20], [20], [20], [21], [28], focus narrowly on specific building types, or optimize models under constrained environmental assumptions. Additionally, the influence of dataset diversity on model performance has not been systematically investigated. To address these shortcomings, this research conducts a comprehensive experimental evaluation using a dataset that incorporates a wide range of building parameters. By analyzing how factors such as insulation levels, window types, building orientation, materials, and occupancy patterns impact predictive accuracy, this study aims to improve model robustness and offer practical insights for enhancing residential energy efficiency in Egypt.

Table 1. A comprehensive review of the literature on predicting energy consumption.

Category	Reference	Dataset	Building Type	Location	Sample Size	Model	Prediction
Prediction of Heating and Cooling Loads Using Classical Machine Learning Models	[20]	Tsanas and Xifara [19]	Residential	-	768 instances	RF	HL and CL
	[21]	Tsanas and Xifara [19]	Residential	-	768 instances	XGBoost	HL and CL
	[22]	Tsanas and Xifara [19]	Residential	-	768 instances	LR	HL and CL
	[23]	-	Residential	-	-	ETTO (Extra Tree Regression + Tyrannosaurus Rex Optimization)	HL
Hybrid and Ensemble Approaches for Load Prediction	[24]	-	Residential	Athens	768 instances	XACM (XGBoost + AdaBoost + CMAES)	HL and CL
	[25]	Tsanas and Xifara [19]	Residential	-	768 instances	PSOGWO (MLP + PSO + GWO)	HL and CL
Broader Energy Consumption Prediction	[26]	Clayton Miller and Forrest Meggers [27]	Non-residential	New York	507 instances	SVR, ANN, Ensemble	Hourly building energy consumption
	[28]	-	Residential	United Kingdom	5,000 instances	ANN, GB, KNN, DNN, RF, DT, SVM, LR, Stacking	Annual building energy consumption
Feature Selection, Causal Analysis, and Online Learning	[30]	Tsanas and Xifara [19]	Residential	-	768 instances	ELM, OSELM	HL and CL
	[29]	Pavel Zorin and Oleg Stukach [7]	Residential	Tomsk, Russia	-	Stacking	Heating energy consumption
	[31]	-	Non-residential	China	2 instances	Stacking, RF, XGBoost, GBDT, SVM, KNN	HL

3. Dataset and Preprocessing

In this section, we first discuss the dataset used for building energy performance prediction. Then, we outline the steps taken to prepare the data for analysis.

3.1. Dataset Description

This study uses a dataset collected through the simulation of multiple buildings generated with the DesignBuilder software [32]. DesignBuilder [32] is an environmental analysis tool that integrates with general Building Information Modeling (BIM) platforms such as Autodesk Revit Architecture [33]. It enables detailed preliminary analyses of building energy demand and performance through a wide range of analytical functions and an interactive visual interface that allows users to visualize results directly within the building model.

To construct the dataset, HL and CL values were recorded for 640 simulated residential buildings. These parameters, hereafter referred to as the output variables and denoted as y, were determined based on varying building designs and configurations. Table 2 presents the mathematical

representation of each variable, the number of possible values, and descriptions of the input and output variables used.

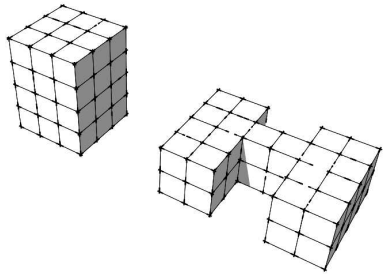
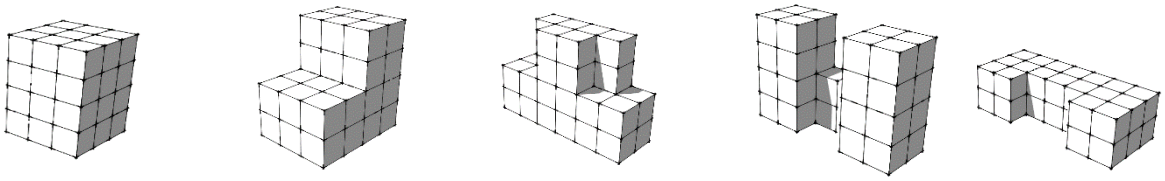


Figure 1. Thirty-six cubic elements are utilized to create the simulated building shapes.

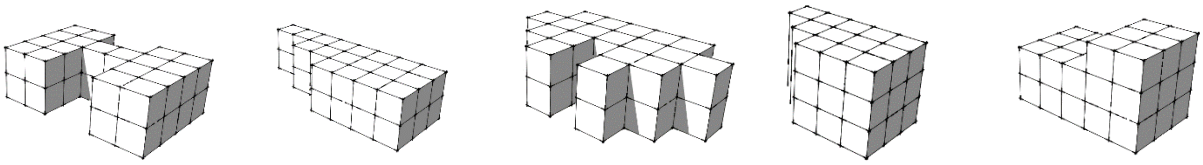
The building geometries are based on a modular system using an elementary cube (3.5 × 3.5 × 3.5 m). Thirty-six cubes were employed (Figure 1) to create different building shapes. Twenty unique forms (Figure 2) were selected for simulation, designed to reflect varying degrees of relative compactness (RC), an indicator of building shape efficiency. RC is calculated according to:

$$RC = \frac{6 \times V^{2/3}}{A} \quad (1)$$

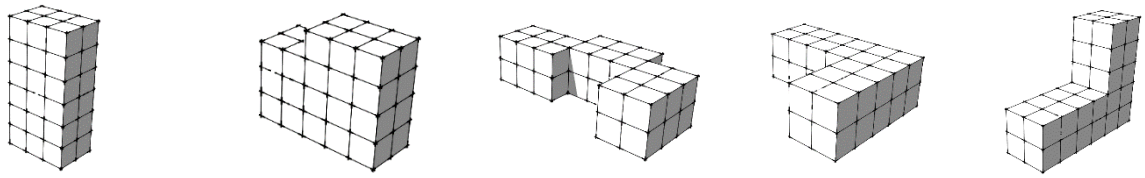
where A denotes the building's surface area, and V refers to its volume.



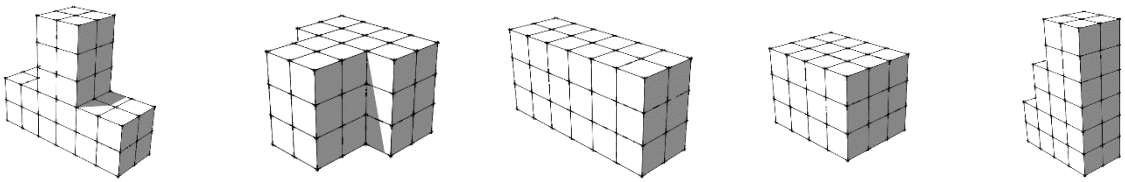
a) Form 1 RC = 0.99 | b) Form 2 RC = 0.91 | c) Form 3 RC = 0.82 | d) Form 4 RC = 0.74 | e) Form 5 RC = 0.82



f) Form 6 RC = 0.74 | g) Form 7 RC = 0.82 | h) Form 8 RC = 0.74 | i) Form 9 RC = 0.91 | j) Form 10 RC = 0.91



k) Form 11 RC = 0.91 | l) Form 12 RC = 0.91 | m) Form 13 RC = 0.78 | n) Form 14 RC = 0.82 | o) Form 15 RC = 0.82



p) Form 16	q) Form 17	r) Form 18	s) Form 19	t) Form 20
RC = 0.82	RC = 0.91	RC = 0.91	RC = 0.99	RC =0.86

Figure 2. Used forms with Relative Compactness Coefficient

The simulations also define detailed building areas, including roof area, floor area, wall area, and overall height (Figure 3). Orientation is considered along the four cardinal directions (North, East, South, and West). For glazing, three glazing-to-floor area ratios (40%, 25%, and 10%) were tested. Additionally, four distinct glazing distribution models were simulated:

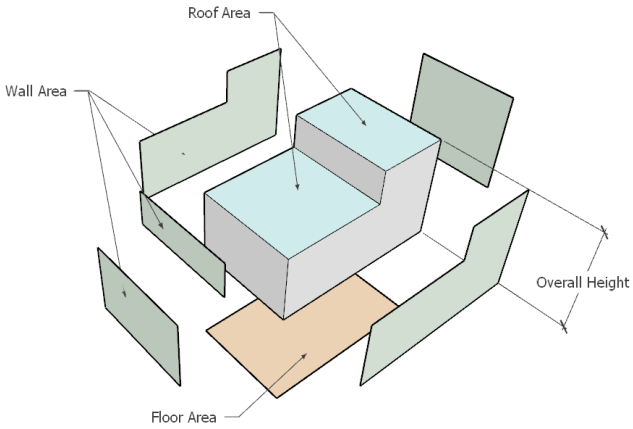


Figure 3. Definition of Building Area Space

- North-focused: 55% glazing on the north side, 15% on each of the others
- East-focused: 55% glazing on the east side, 15% on the others
- South-focused: 55% glazing on the south side, 15% on the others
- West-focused: 55% glazing on the west side, 15% on the others

Scenarios with no glazing were also included. Each building was rotated to face all four orientations. Based on this setup, the experimental dataset comprises ($20 \times 7 \times 4 + 20 \times 4 = 640$) samples of different building configurations. Further details on input and output parameters are shown in Table 2.

Building materials were chosen based on standard construction practices and low U-values to reflect realistic conditions. Specifically, the following components were modeled:

- Walls: 0.403 W/m²K
- Floors: 4.730 W/m²K
- Roofs: 0.250 W/m²K

Simulations were conducted assuming a residential setting located in Cairo, Egypt, with an occupancy of four persons per cube. Internal conditions were set as follows: clothing insulation level of 0.6 clo, relative humidity of 60%, air speed of 0.30 m/s, and lighting level of 300 Lux. Sensible and latent internal gains were set to 5 and 2 W/m², respectively. The air infiltration rate was fixed at 0.5 air changes per hour, with a wind sensitivity factor of 0.25.

For thermal operation, a mixed-mode system was modeled with 95% efficiency and a thermostat range of 19°C to 24°C. Building systems were assumed to operate for 15–20 hours on weekdays and 10–20 hours on weekends.

Table 2. Dataset Input and output variables

Mathematical Representation	Input/ output variable	Number of possible values
X1	Relative Compactness	6
X2	Surface Area	7
X3	Wall Area	8
X4	Roof Area	7
X5	Overall height	5
X6	Orientation	4
X7	Window to wall ratio	25
X8	Glazing Area Distribution	5
Y1	Heating Load	640
Y2	Cooling Load	640

3.2. Data Preprocessing

Data preprocessing is a vital step in preparing the collected data for analysis or model training, as it helps improve data quality and ensures reliable results [34]. Hence, this section describes the steps taken to prepare the dataset for ML model training.

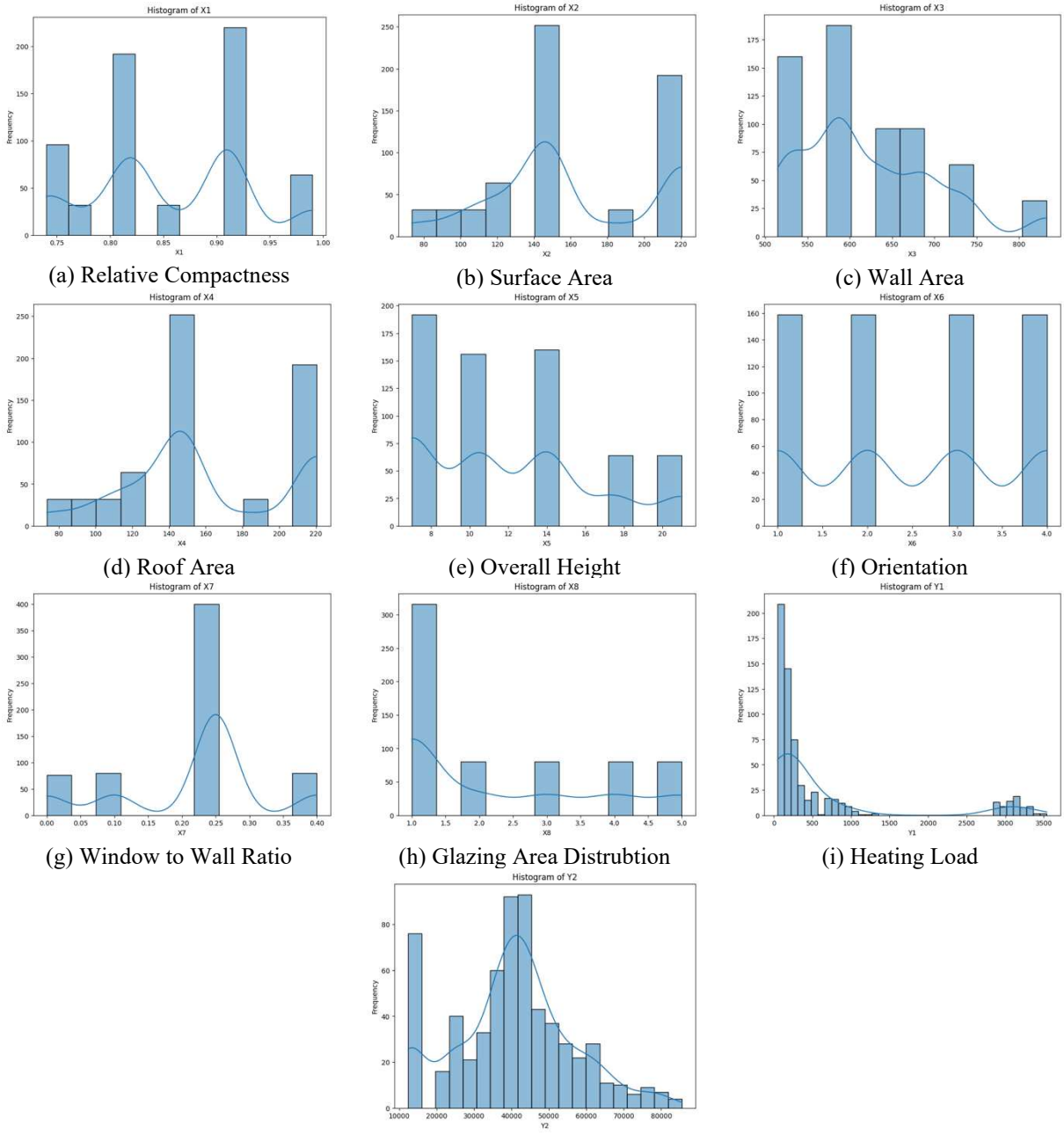
The dataset consists of 640 samples, each representing a distinct building configuration. Every sample contained eight input features describing the building geometry and fenestration properties. The output variables (targets) are Heating Load and Cooling Load. Heating load refers to the amount of heating energy required, measured in kilowatt-hours (kWh), while cooling load is the amount of cooling energy required, measured in kilowatt-hours (kWh).

All features and target variables are verified to be numerical and continuous, ensuring compatibility with machine-learning regression algorithms. The input features consist of a combination of continuous variables (such as Relative Compactness and Surface Area) and categorical variables encoded numerically (such as Orientation and Glazing Area Distribution). This mixture reflects realistic building characteristics affecting thermal performance.

3.2.1. Feature Distributions

A descriptive data analysis is conducted to examine the distribution, relationships, and potential multicollinearity among the features prior to model training. The distributions of all input features and target variables are visualized using histograms [35], [36], as illustrated in Figure 4. These histograms provide insights into variability, frequency, and potential skewness of the data.

As the figure shows, categorical Features such as orientation and glazing area exhibit discrete distributions, with certain values occurring more frequently. For instance, orientation shows four distinct peaks corresponding to predefined categories. Alternatively, relative compactness, surface area, and overall height display a multimodal distribution, indicating the presence of distinct groups within the dataset. The peaks suggest specific building configurations that occur more frequently. Also, the distributions of wall area and roof area are irregular, with certain values appearing significantly more often than others, potentially due to design constraints in the dataset.



(a) Relative Compactness (b) Surface Area (c) Wall Area

(d) Roof Area (e) Overall Height (f) Orientation

(g) Window to Wall Ratio (h) Glazing Area Distrubtion (i) Heating Load

(j) Cooling Load

Figure 4. The Histogram Visualization of All Input Features and Target Variables Distribution

For the target variables, heating load is highly skewed, with most values concentrated at the lower end of the scale, suggesting that the majority of buildings have relatively low heating requirements. By comparison, cooling load demonstrates a more dispersed distribution, reflecting greater variability in cooling energy demand. The unbalanced Heating Load distribution highlights the need for a logarithmic transformation to improve distributional symmetry. For consistency, the same transformation was also applied to cooling load. As shown in Figure 5, the transformation effectively reduced skewness and resulted in more symmetrical distributions.

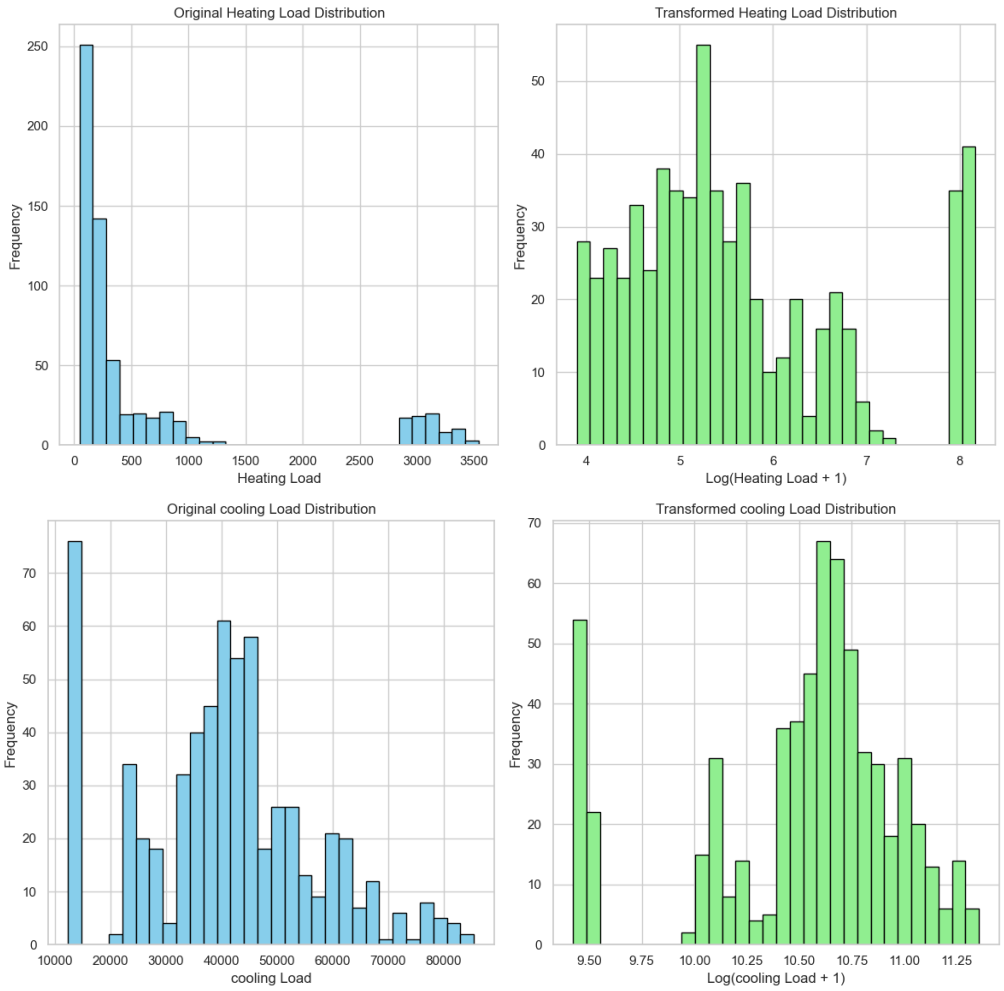


Figure 5. The Histogram Visualization of Transformed Target Variables Distribution

3.2.2. Correlation Analysis

A correlation analysis was conducted to examine the strength and direction of linear relationships between the input features and the target variables. The results are visualized in the form of a heatmap [37], [38], as illustrated in Figure 6. A heatmap is a graphical representation of data where individual values contained in a matrix are depicted as colors. In the context of correlation analysis, a heatmap uses a color gradient to indicate the strength of the correlation between pairs of variables. Values close to 1 (dark red) represent strong positive correlations, meaning that as one variable increases, the other tends to increase. Alternatively, values close to -1 (dark blue) indicate strong negative correlations, where one variable tends to decrease as the other increases. Values near 0 (light colors) imply little or no linear relationship [37].

The results indicated strong correlations between these target variables and some features, suggesting that these attributes play a significant role in determining energy efficiency. For example, the Window-to-Wall Ratio shows a strong negative correlation with Heating Load (−0.83), indicating that increasing window size generally decreases the need for heating. Cooling Load also correlates negatively with Heating Load (−0.75), suggesting that buildings requiring less heating tend to require more cooling energy, reflecting typical energy behavior in temperate climates.

Glazing Area Distribution exhibits a moderate negative correlation with Heating Load (−0.36). In contrast, features such as Surface Area, Roof Area, Orientation, and Wall Area have a weak or negligible direct correlation with the target variables individually. These findings imply that certain features, particularly those related to building fenestration and envelope design, have a more substantial influence on energy performance compared to purely geometric attributes.

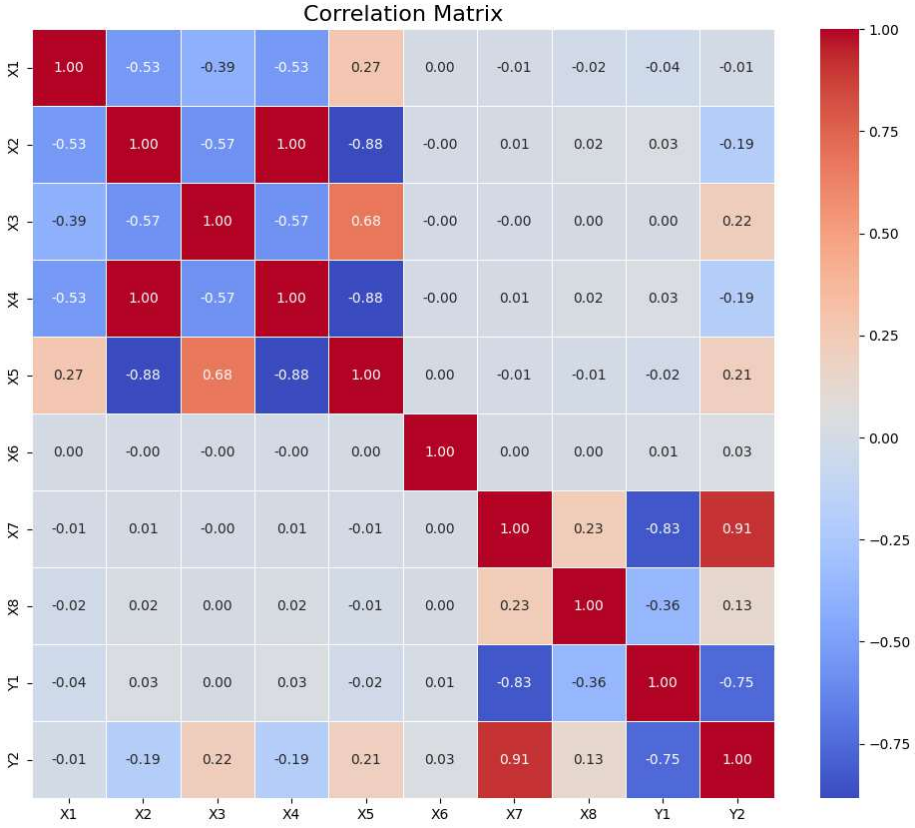


Figure 6. Correlation Heatmap of Input Features and Target Variables

3.2.3. Multicollinearity Assessment

Multicollinearity refers to the phenomenon where two or more input features are highly linearly correlated with each other [39]. When multicollinearity is present, it can distort the importance of individual features in regression models, lead to unstable parameter estimates, and reduce the interpretability of the model. In this research, multicollinearity was assessed using the Pearson correlation matrix presented in Figure 6. The heatmap revealed several vital relationships. First, Surface Area and Roof Area are found to be perfectly correlated. This strong linear dependence arises from the modular design of the simulated buildings, where variations in surface area directly influence the roof area. Second, Surface Area showed a high correlation with Wall Area, although the relationship was slightly weaker compared to Roof Area. These observations indicate that the dataset contains features that are not fully independent. The presence of such multicollinearity can be problematic for specific machine learning algorithms, especially linear regression and support vector regression. It may inflate the variance of model coefficients and lead to overfitting or unstable predictions.

3.3. Feature Engineering

Following the descriptive analysis and multicollinearity assessment, feature engineering is conducted to enhance the predictive quality of the machine learning models and address issues identified in the dataset [40]. Given the presence of multicollinearity between certain geometric features, particularly between Surface Area, Roof Area, and Wall Area, a decision was made to select input features carefully to avoid redundancy. Instead of removing features manually, feature selection strategies are applied systematically during model training. Specifically, recursive feature elimination (RFE) [41] or feature importance rankings based on tree-based models [42] can help identify the most influential features for Heating and Cooling Load prediction. Therefore, in this study, all available features were initially retained to allow the models themselves to determine the relevance of each attribute.

As for encoding categorical variables, the dataset included two features that were originally categorical but numerically encoded as follows: Orientation (values 1–4), and Glazing Area Distribution (values 0–4). Since these features are already encoded as integers representing discrete categories, no additional transformation (such as one-hot encoding [43]) was applied. The models employed are capable of handling integer-coded categorical variables directly without loss of information.

Finally, before model training, feature scaling [44] was applied to all input variables to ensure numerical stability and to prevent features with larger scales from dominating others. Standardization was used, transforming each feature to have a mean of zero and a standard deviation of one. The scaling was performed using the StandardScaler from the Scikit-learn library, according to the transformation [45]:

$$z = \frac{x-\mu}{\sigma} \quad (2)$$

where x is the original feature value, μ is the mean of the feature, and σ is the standard deviation. This standardization process is crucial for models such as Support Vector Regression (SVR) and k-nearest Neighbors (k-NN), which are sensitive to the magnitude of input features. Following feature engineering, the dataset became suitable for supervised learning models aimed at predicting the energy demands of residential buildings.

4. Machine Learning Models

This section presents the machine learning models applied to predict the heating and cooling loads of residential buildings. A combination of ensemble methods, classical regressors, and regularized regression models was selected to evaluate the performance across different learning paradigms. Each model is briefly introduced below.

4.1. Selected Algorithms

In this study, a diverse set of machine learning algorithms was selected to predict the heating and cooling loads of residential buildings. The choice of models was motivated by the need to assess a wide range of learning paradigms, balance between bias and variance, and evaluate models with varying levels of complexity and interpretability. A summary of the selected models is summarized in Table 3. As the table shows, the selected algorithms can be categorized into ensemble methods [46], non-parametric models [47], kernel-based methods [48], and linear models, each offering distinct advantages for the problem at hand.

First, several ensemble techniques were selected, including the Gradient Boosting Regressor [49], Random Forest Regressor [50], Bagging Regressor [51], and AdaBoost Regressor [52]. Ensemble methods combine multiple base learners to improve predictive performance and robustness. These

models [46], [53] are particularly well-suited for capturing non-linear relationships and interactions among building design variables.

Second, tree-based and instance-based models [54] include the Decision Tree Regressor [54] and K-Neighbors Regressor [55]. These models were included to provide benchmarks for simpler, interpretable models. Decision Tree Regressor offers a straightforward, hierarchical partitioning of the feature space and serves as a baseline for evaluating the benefits of ensemble enhancements [54]. Similarly, the K-Neighbors Regressor is a non-parametric method that predicts outcomes based on the local structure of the data, making no strong assumptions about the underlying data distribution. Together, these models allow an examination of the performance of basic learners compared to more sophisticated ensemble methods [55].

Third, the Support Vector Regressor [48] was selected to represent kernel-based learning. SVR effectively models complex, non-linear patterns by mapping input features into higher-dimensional spaces using kernel functions. Its ability to control model complexity through regularization parameters makes it a valuable addition to the comparison. SVR is particularly relevant for datasets where the relationship between inputs and outputs is not purely linear, as is often the case in building energy performance.

Finally, Linear Regression [56] and Lasso Regression [57] were selected to represent classical and regularized linear approaches. Linear Regression [56] provides a simple and interpretable model that serves as a baseline for understanding the benefit of introducing non-linearity and ensemble strategies. Lasso Regression [57] introduces an L1 penalty that encourages sparsity in the feature coefficients, helping to address potential multicollinearity and overfitting. Including linear models allows the study to evaluate whether simple models are sufficient for predicting energy loads or whether more complex methods are necessary.

Table 3. Summary of Selected Machine Learning Models

Category	**Model**	**Main Characteristics**
Ensemble Method	Gradient Boosting Regressor	Sequentially reduces prediction errors using gradient descent
	Random Forest Regressor	Aggregates multiple decision trees trained on random subsets
	Bagging Regressor	Reduces variance by averaging predictions from bootstrap samples
	AdaBoost Regressor	Focuses on difficult-to-predict instances across iterations
Tree-Based and Instance-Based Models	Decision Tree Regressor	Hierarchical data partitioning; prone to overfitting alone
	K-Neighbors Regressor	Predicts based on the average of nearest neighbors
Kernel-Based Method	Support Vector Regressor	Captures non-linear patterns through kernel transformations
Linear Model	Linear Regression	Models a linear relationship between inputs and outputs
	Lasso Regression	Performs feature selection via L1 regularization

4.1.1. Gradient Boosting Regressor

The Gradient Boosting Regressor (GBR) [49] is an ensemble learning technique that constructs a strong predictive model by sequentially combining multiple weak learners, typically decision trees. Unlike methods where models are trained independently, Gradient Boosting builds each new model to correct the errors made by the ensemble thus far. This sequential learning process allows the algorithm to progressively improve its predictions, focusing more attention on instances that previous models handled poorly.

Figure 7 illustrates the general workflow of Gradient Boosting. As the figure shows, the training process begins with an initial prediction, which is often the mean value of the target variable in regression tasks. Subsequently, at each iteration, a new decision tree is trained to predict the residuals — the differences between the actual target values and the current predictions. These residuals represent the areas where the model is underperforming and needs improvement. After training, the predictions from the new tree are added to the ensemble's overall prediction, adjusted by a learning rate parameter that controls how much influence each new tree exerts.

One of the principal advantages of Gradient Boosting is its ability to achieve high predictive accuracy, particularly on structured datasets with complex, non-linear relationships among variables. The method is highly flexible, capable of optimizing a variety of loss functions, and can automatically capture feature interactions without requiring extensive feature engineering.

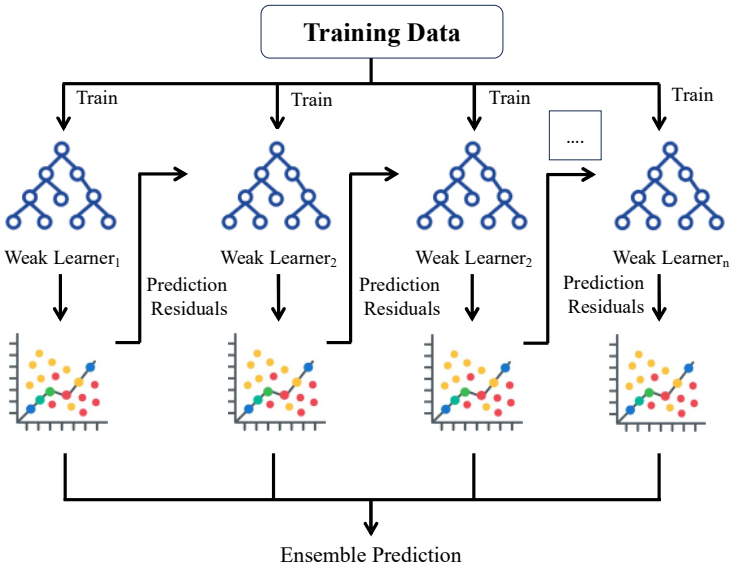


Figure 7. General Workflow of Gradient Boosting

4.1.2. Random Forest Regressor

The Random Forest Regressor (RFR) [50] is an ensemble learning method designed to improve predictive accuracy and control overfitting by constructing a large collection of decision trees and aggregating their outputs. It belongs to the family of bagging algorithms [58], where multiple base learners are trained independently, and their predictions are combined, typically by averaging in the case of regression tasks.

The training of a Random Forest begins by generating multiple bootstrap samples from the original training data. Each sample is created by randomly selecting observations with replacements, ensuring that different trees are exposed to slightly different subsets of the data. For every tree in the forest, a random subset of features is also selected at each split, introducing an additional layer

of randomness. This feature randomness ensures that the trees are de-correlated, which improves the overall generalization performance of the model. The general structure of a Random Forest is illustrated in Figure 8, where multiple independent decision trees contribute collectively to the final prediction.

Once trained, each decision tree in the ensemble produces its own prediction. The final output of the Random Forest Regressor is obtained by averaging the predictions of all individual trees. This averaging process significantly reduces variance compared to a single decision tree, leading to more stable and reliable predictions even when dealing with noisy or complex datasets.

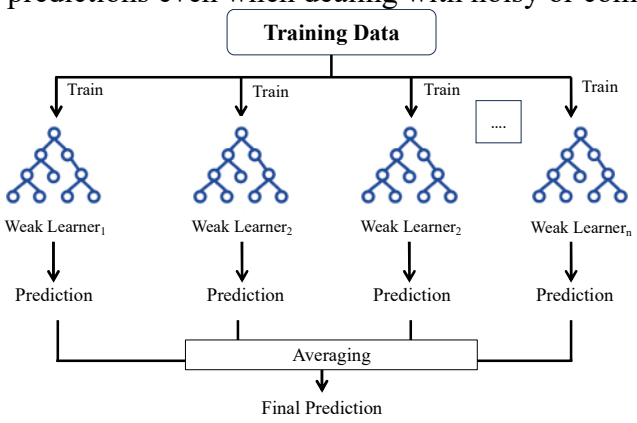


Figure 8. General Structure of Random Forest Regressor

4.1.3. Decision Tree Regressor

The Decision Tree Regressor (DTR) [50] is a supervised learning method used for predicting continuous target variables by learning simple decision rules inferred from data features. It builds a tree-like model of decisions, where the data is recursively partitioned into subsets based on feature values that result in the greatest reduction of prediction error. Each internal node of the tree represents a decision on a feature, each branch represents an outcome of that decision, and each leaf node represents a final prediction.

The process of training a Decision Tree begins by evaluating all possible splits across all features and selecting the feature and threshold that best divides the data to minimize a chosen impurity metric, such as Mean Squared Error (MSE) [59] for regression tasks. This partitioning is applied recursively, creating increasingly refined subsets of the data at each level of the tree. The objective is to generate subsets that are as homogeneous as possible concerning the target variable. An example of the structure of a Decision Tree is depicted in Figure 9, illustrating how data is split at each node to form predictions at the leaves progressively.

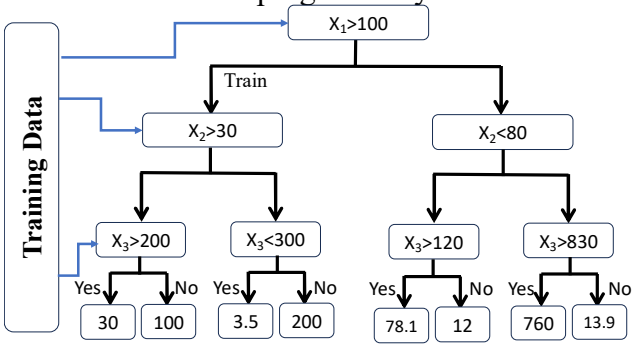


Figure 9. General Structure of Decision Tree

One of the main advantages of Decision Trees is their simplicity and interpretability. The hierarchical structure provides a transparent view of how decisions are made, which can be easily visualized and understood. Decision Trees are also capable of modeling non-linear relationships and interactions between features without requiring feature scaling or transformation.

4.1.4. Bagging Regressor

The Bagging Regressor [60] is an ensemble learning method that aims to improve the stability and accuracy of machine learning algorithms by reducing variance through the technique of bootstrap aggregation, commonly referred to as "bagging." The fundamental idea behind bagging is to create multiple versions of a predictor and use them to generate an aggregated prediction. In the case of regression tasks, this aggregation is typically performed by averaging the individual predictions [61].

The training process for the Bagging Regressor begins by creating multiple bootstrap samples from the original training data. Each bootstrap sample is obtained by randomly selecting observations from the dataset with replacement, allowing some samples to be repeated while others may be omitted. A base regressor, often a decision tree, is then trained independently on each of these bootstrap samples. Since each base model sees a different subset of the data, they are likely to produce slightly different models, each capturing different aspects of the training set.

Once all base models have been trained, the Bagging Regressor combines their predictions by averaging them, thus producing the final output. This averaging process helps in reducing the model variance, which is particularly beneficial for high-variance models like decision trees. As a result, the Bagging Regressor typically achieves better generalization performance on unseen data compared to a single model trained on the entire dataset. The general approach of the Bagging Regressor is illustrated in Figure 10, where multiple independent base models trained on different bootstrap samples are combined to produce a single, averaged prediction.

Bagging is particularly effective because it reduces the impact of noise and anomalies present in the training data. By averaging over multiple models, it smooths out predictions and leads to a more robust and stable overall model. Furthermore, because each model is trained independently, bagging is highly parallelizable, making it computationally efficient on modern hardware.

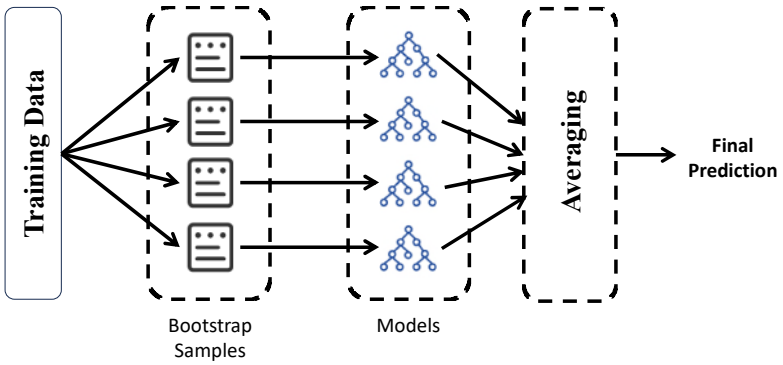


Figure 10. General Workflow of Bagging Regressor

4.1.5. AdaBoost Regressor

The AdaBoost Regressor (Adaptive Boosting) is an ensemble learning method [52] designed to improve predictive performance by combining multiple weak learners into a single strong learner

through an iterative, weighted training process. Unlike bagging techniques, which train base learners independently, AdaBoost trains each model sequentially, where each new model focuses more on the instances that previous models predicted poorly.
The training process begins by assigning equal weights to all training instances. After the first base regressor is trained, the model's prediction errors are analyzed. Samples that were predicted incorrectly are assigned higher weights, while correctly predicted samples receive lower weights. This reweighting process ensures that subsequent base learners pay more attention to the difficult instances, allowing the ensemble to progressively correct its own mistakes. At each iteration, a new base regressor, typically a shallow decision tree, is trained on the weighted dataset. The performance of each learner is evaluated, and a corresponding weight is assigned to its prediction based on its accuracy. The final prediction of the AdaBoost Regressor is then computed as a weighted sum of the predictions from all individual models. Therefore, the model sequentially fits learners on reweighted data, focusing increasingly on hard-to-predict instances, and combines their outputs into a final, strong predictor.
One of the primary advantages of AdaBoost is its ability to improve the performance of weak learners that are only slightly better than random guessing. By focusing learning effort on difficult cases, AdaBoost often achieves higher accuracy than bagging methods when the base learners are simple. It is particularly effective in handling moderately complex relationships without substantial risk of overfitting, provided that the base learners themselves are not overly complex.

4.1.6. K-Neighbors Regressor

The K-Neighbors Regressor (KNNR) [55] is a non-parametric method used for regression tasks that predicts the target value of a new sample based on the target values of its k closest neighbors in the feature space. Rather than learning an explicit mapping from inputs to outputs during training, KNNR retains the entire training dataset and performs all computations at prediction time. When a prediction is required, the algorithm identifies the k samples in the training data that are nearest to the query point according to a specified distance metric, typically Euclidean distance. The predicted output is then computed as the average (or, in some variants, the weighted average) of the target values associated with these neighbors. By relying on local patterns in the data, KNNR is able to adapt naturally to non-linear relationships without the need for complex model fitting.
One of the primary advantages of the K-Neighbors Regressor is its simplicity and intuitive approach. It makes very few assumptions about the underlying data distribution, making it a flexible choice for a wide range of regression problems. Additionally, KNNR can perform well when the data is well-clustered and when local relationships are strong.

4.1.7. Support Vector Regressor

The Support Vector regression [48] is an extension of support vector machines (SVM) designed for regression tasks. Unlike traditional regression models, which aim to minimize the prediction error directly, SVR attempts to find a function that approximates the target values within a specified margin of tolerance, denoted by ε (epsilon), while maintaining the model as flat as possible.
The core idea of SVR is to ignore errors that fall within this ε-tube around the true target values. Predictions that lie within this tube are considered sufficiently accurate and do not contribute to the loss. Only deviations larger than ε are penalized. This formulation leads to a sparse solution

where only a subset of the training points, known as support vectors, influence the model's predictions, enhancing computational efficiency and model generalization.
SVR achieves this by solving a constrained optimization problem that balances model complexity and error tolerance. The flatness of the regression function is controlled by minimizing the norm of the coefficients, while the slack variables allow some flexibility for errors that exceed the ε threshold. A regularization parameter, often denoted as C, controls the trade-off between model complexity and the amount by which deviations larger than ε are tolerated. A conceptual illustration of the SVR approach is provided in Figure 11, showing the ε-tube around the regression function and the support vectors that define the model.

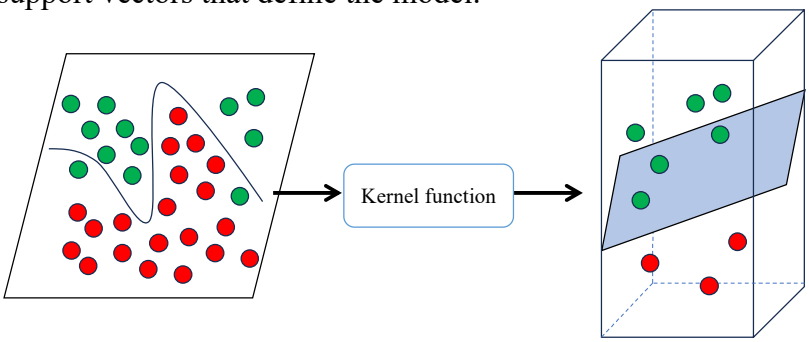


Figure 11. General Workflow of Support Vector Regressor

An important strength of SVR lies in its ability to model non-linear relationships through the use of kernel functions. By applying a kernel trick, SVR maps the input features into a higher-dimensional space where a linear regression model can be fitted. Common kernels include the radial basis function (RBF), polynomial, and linear kernels. The choice of kernel and its parameters significantly affects the model's ability to capture complex patterns.
SVR is particularly effective when the number of features is large relative to the number of samples, and it offers robust performance in the presence of high-dimensional data. However, SVR can be sensitive to the selection of hyperparameters such as C, ε, and kernel parameters, which typically require careful tuning through cross-validation. Additionally, the computational cost can become significant for large datasets, as the complexity of training depends on the number of support vectors.

4.1.8. Linear Regression

Linear Regression [56] is one of the most fundamental and widely used techniques in supervised learning, particularly for regression tasks. The objective of linear regression is to model the relationship between a set of input features and a continuous target variable by fitting a linear equation to the observed data. The model assumes that the target variable can be expressed as a weighted sum of the input features, plus an intercept term.
Training a linear regression model involves estimating the coefficients, or weights, for each input feature such that the sum of the squared differences between the actual target values and the predicted values is minimized. This process, known as Ordinary Least Squares (OLS) [56], seeks to find the line or hyperplane that best fits the data in a manner that minimizes prediction error. The resulting model provides a straightforward interpretation: each coefficient represents the expected change in the target variable for a one-unit change in the corresponding feature, assuming all other features remain constant.

One of the primary advantages of linear regression is its simplicity and interpretability. The model's assumptions are easy to understand, and the fitted coefficients offer direct insights into the influence of each feature on the target variable. Furthermore, linear regression models are computationally efficient to train, even on relatively large datasets.
However, linear regression comes with several limitations. It assumes a linear relationship between the inputs and the output, which may not hold in complex real-world scenarios. The model is also sensitive to multicollinearity among input features, where highly correlated predictors can distort the estimated coefficients and reduce model reliability. Additionally, linear regression is sensitive to outliers, which can disproportionately influence the fitted model [56], [62].
Despite these limitations, linear regression serves as a critical baseline in predictive modeling. It establishes a benchmark against which more complex models can be compared and often provides surprisingly strong performance when the true relationship between inputs and outputs is approximately linear.

4.1.9. Lasso Regression

Lasso Regression (Least Absolute Shrinkage and Selection Operator) [57] is a regularized extension of linear regression that addresses some of the limitations of the standard linear model, particularly in situations where multicollinearity among features is present or when feature selection is desired. The primary innovation in Lasso Regression is the addition of an L1 penalty term to the loss function, which encourages sparsity in the coefficient estimates.
In Lasso Regression, the model seeks to minimize the residual sum of squares while simultaneously constraining the sum of the absolute values of the coefficients. This regularization term imposes a constraint that tends to shrink some coefficients exactly to zero, effectively removing the corresponding features from the model. As a result, Lasso not only helps prevent overfitting by reducing model complexity but also performs automatic feature selection, retaining only the most relevant predictors.
One of the main advantages of Lasso Regression is its ability to handle high-dimensional datasets where the number of features may be large relative to the number of observations. By eliminating irrelevant or redundant features, Lasso simplifies the model, making it more interpretable and often improving its generalization performance.
However, Lasso has certain limitations. When there are highly correlated predictors, Lasso may arbitrarily select one variable from the group while shrinking the others to zero, potentially losing valuable information. Moreover, suppose the number of predictors greatly exceeds the number of observations. In that case, Lasso may select at most as many predictors as there are observations, which can be restrictive in some applications [57].

4.2. Model Training

Each of the selected machine learning models is trained and evaluated on the standardized dataset for predicting heating and cooling loads described in Section 3.1. The training process emphasizes consistency across the models to ensure a fair comparison while also accounting for each learning algorithm's specific requirements.
First, the dataset was partitioned into training and testing subsets to assess model generalization. An 80/20 split was used, with 80% of the samples allocated to the training set and the remaining 20% reserved for testing. The split was performed randomly but in a stratified manner, ensuring that the distribution of the target variables — heating load and cooling load — was preserved

across both subsets. This approach minimizes sampling bias and ensures that both the training and testing sets are representative of the overall dataset. As discussed in Section 3.2, feature scaling was applied to the input variables using standardization. The StandardScaler method [45] was employed to transform the features to have zero mean and unit variance. This step is particularly important for models such as Support Vector Regressor and K-Neighbors Regressor, which are sensitive to the magnitude of feature values. The same scaling parameters computed on the training set were applied to the test set to avoid information leakage.

Each machine learning model was trained on the standardized training data according to its specific requirements. Default hyperparameters were initially used to establish baseline performance. Subsequently, for models where initial performance was suboptimal, manual tuning of key hyperparameters was conducted based on cross-validation results.

For example, in ensemble models like Gradient Boosting and Random Forest, the number of estimators (trees), maximum depth, and learning rates were adjusted. For SVR, the penalty parameters C, epsilon, and kernel types were fine-tuned. Similarly, the number of neighbors k for the K-Neighbors Regressor was selected based on validation error minimization.

The training was performed independently for the two prediction tasks — heating load and cooling load — to allow the models to optimize separately for each target. This separation acknowledges that the two tasks, while related, may have different underlying predictive structures and feature importances. Also, early stopping techniques were considered for boosting methods where applicable to prevent overfitting. In addition, regularization parameters in Lasso Regression were tuned using internal validation to ensure optimal feature selection without excessive shrinkage. All models were trained using the Scikit-learn library in Python [63] , utilizing consistent random seeds to ensure the reproducibility of the results.

4.3. Performance Metrics

The performance metrics used to evaluate the proposed energy prediction model are statistical measures that determine how well the model fits the data. These key measures are the root mean square error (RMSE); the adjusted R^2 refines the conventional R^2 (coefficient of determination) by imposing a penalty for the inclusion of superfluous predictors; the mean absolute error (MAE); and the mean absolute percentage error (MAPE) [64]. These metrics provide insights into the model's accuracy, precision, and overall fit.

A model will be considered to fit the data well when all error measures are low values, indicating that the differences between the actual values and the model's predicted values are small and unbiased. A higher Adjusted R-squared value indicates that the model correctly fits the data. Fundamentally, these error metrics assess the model's ability to anticipate energy demands using the error values. The formulas utilized to calculate these performance measures are as follows [64]

$$\mathrm{RMSE} = \sqrt{{}^{1}/_{S}\sum_{i=1}^{S}(y_i - \acute{y}_i)^2} \tag{3}$$

$$R^2 = 1 - \frac{\sum_{i=1}^{S}(y_i - \acute{y}_i)^2}{\sum_{i=1}^{S}(y_i - \bar{y}_m)^2} \tag{4}$$

$$\text{Adjusted } R^2 = 1 - [(1 - R^2) * (s - 1) / (s - p - 1)] \tag{5}$$

$$\mathrm{MAE} = {}^{1}/_{\mathrm{S}} \sum_{\mathrm{i}=1}^{\mathrm{S}} |\mathrm{y_i} - \acute{\mathrm{y}}_\mathrm{i}| \tag{6}$$

$$\mathrm{MAPE} = \frac{1}{\mathrm{S}} \times \sum_{\mathrm{i}=1}^{\mathrm{S}} \frac{|\mathrm{y_i} - \acute{\mathrm{y}}_\mathrm{i}|}{\mathrm{y_i}} \times 100\% \tag{7}$$

The actual HL and CL value responses are denoted by y_i, the predicted value of the HL and CL responses is represented by $\acute{y}_i$, S is the number of samples, p number of predictors and $\bar{y}_m$ is the mean of the actual values.

5. Results and Discussion

Numerical experimental results were recorded using the Python programming language. In this study, the analysis of experimental results revealed significant findings between the selected ML models. According to the performance evaluation criteria, a model that achieves values closer to zero for MAE, MSE, and RMSE is considered a good predictive model. MAPE compares the predicted values to the actual values and expresses the accuracy as a percentage. Conversely, a model with adjusted R^2 values closer to 1 is deemed to produce the superlative results. Tables 4 and 5 demonstrate the performance indicators for predicting energy in both HL and CL using various ML models.

Table 4. Performance of HL prediction using ML models.

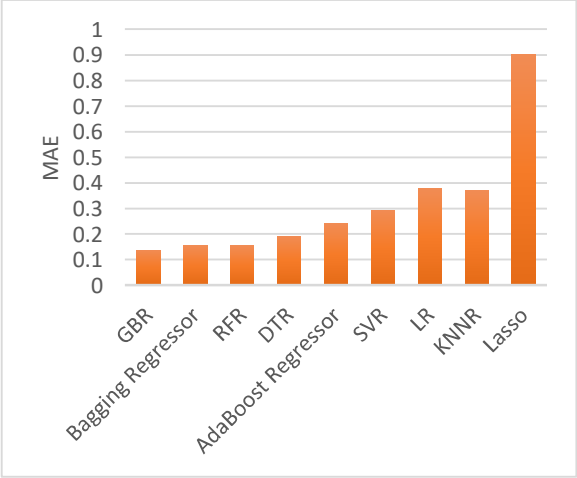
Model	$adjusted\ R^2$	MSE	$RMSE$	MAE	$MAPE(\%)$
Gradient Boosting Regressor (GBR)	0.975	0.035	0.188	0.136	2.6
Bagging Regressor	0.968	0.045	0.211	0.154	3
Random Forest Regressor (RFR)	0.966	0.047	0.216	0.154	3
Decision Tree Regressor (DTR)	0.949	0.071	0.266	0.19	3.8
AdaBoost Regressor	0.932	0.094	0.307	0.239	4.7
Support Vector Regressor (SVR)	0.87	0.18	0.424	0.292	5.7
Linear Regression (LR)	0.837	0.226	0.475	0.377	7.3
K Neighbors Regressor (KNNR)	0.831	0.234	0.483	0.369	6.9
Lasso	0.079	1.276	1.13	0.9	16.1

Table 5. Performance of CL prediction using ML models.

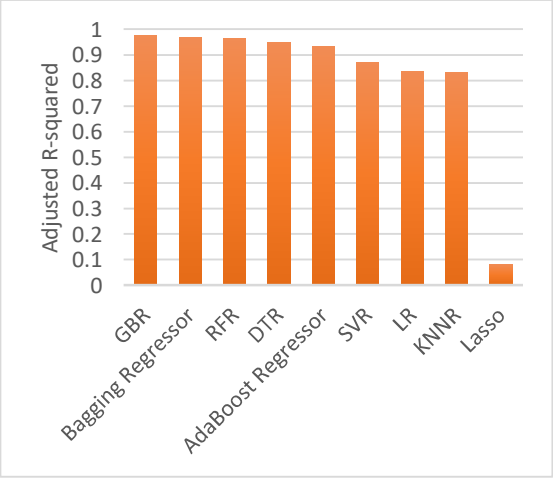
Model	$adjusted\ R^2$	MSE	$RMSE$	MAE	$MAPE(\%)$
Gradient Boosting Regressor (GBR)	0.978	0.005	0.071	0.051	0.5
Bagging Regressor	0.972	0.006	0.08	0.057	0.5
Random Forest Regressor (RFR)	0.967	0.007	0.086	0.061	0.6
Decision Tree Regressor (DTR)	0.953	0.011	0.103	0.069	0.6
AdaBoost Regressor	0.953	0.011	0.103	0.081	0.8
Support Vector Regressor (SVR)	0.934	0.015	0.122	0.096	9
Linear Regression (LR)	0.906	0.021	0.146	0.12	1
K Neighbors Regressor (KNNR)	0.877	0.028	0.166	0.125	1.2
Lasso	-0.016	0.229	0.478	0.361	3.5

For heating load prediction, Gradient Boosting achieved the highest performance with an adjusted R^2 of 0.975. Bagging Regressor (0.968), Random Forest Regressor (0.966), Decision Tree Regressor (0.949), and AdaBoost Regressor (0.932) followed closely, all demonstrating strong predictive capability. In contrast, the Support Vector Regressor showed a lower adjusted R^2 of 0.87, while the K-Neighbors Regressor and Linear Regression achieved only moderate accuracy and exhibited higher error values. Lasso Regression performed poorly, with an adjusted R^2 of 0.079 and substantially higher error metrics. A comparative summary of these results is illustrated in Figure 12.

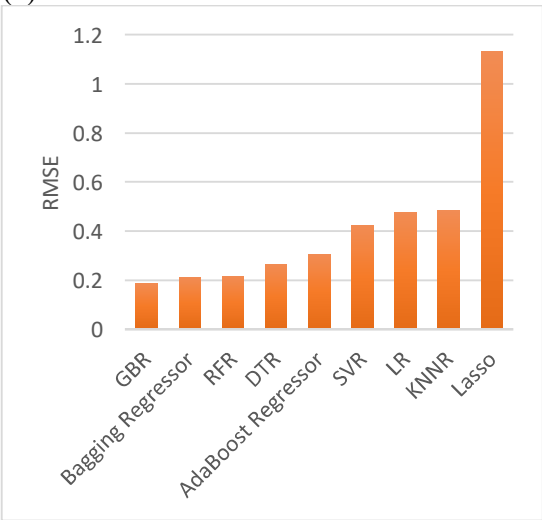
For cooling load prediction, Gradient Boosting achieved the best performance with an adjusted R^2 of 0.978. Bagging Regressor (0.972), Random Forest Regressor (0.967), and both Decision Tree Regressor and AdaBoost Regressor (0.953) also demonstrated strong predictive accuracy, supported by consistently low error values. Support Vector Regressor and Linear Regression provided moderate effectiveness, whereas K-Neighbors Regressor showed reduced accuracy. Lasso Regression yielded a negative adjusted R^2 value (−0.016), confirming its unsuitability for this prediction task. Overall, these results highlight the superiority of ensemble-based models—particularly Gradient Boosting and Bagging—as the most reliable approaches for estimating both heating and cooling loads. Figure 13 presents a comparative summary of these outcomes.



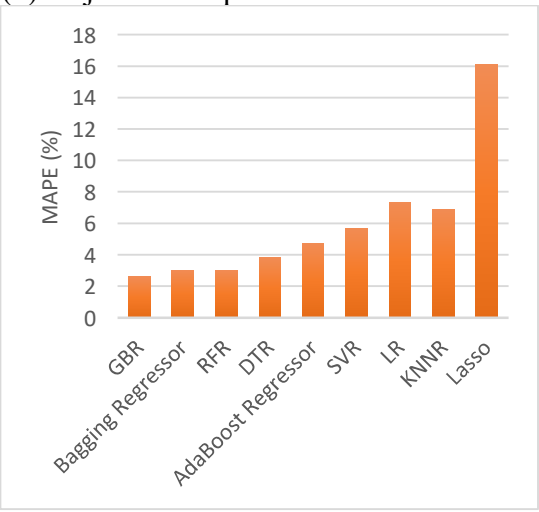
(a) MAE



(b) Adjusted R-squared

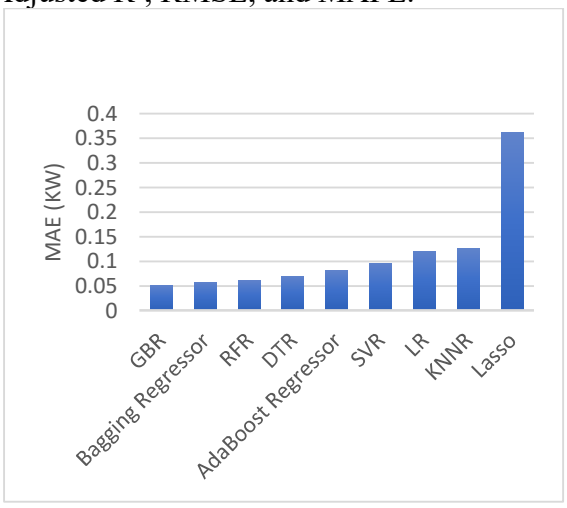


(c) RMSE

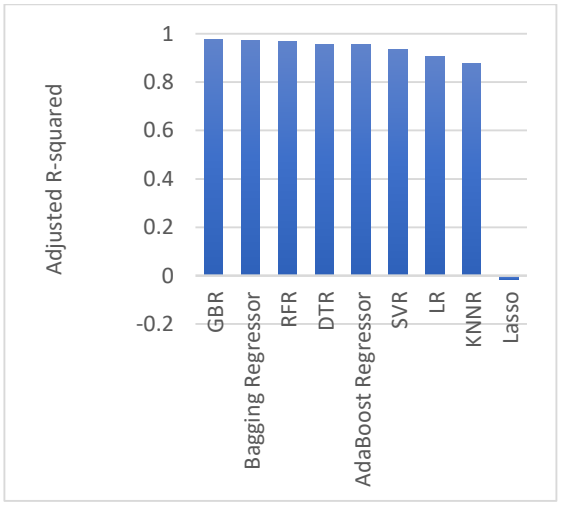


(d) MAPE

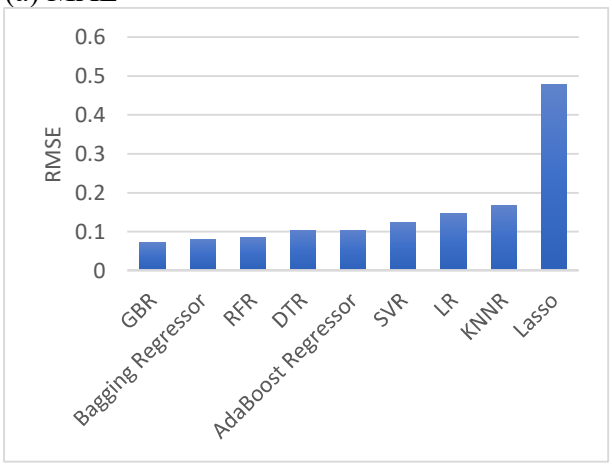
Figure 12. Performance comparison of ML models for Heating Load estimation using MAE, Adjusted R^2, RMSE, and MAPE.



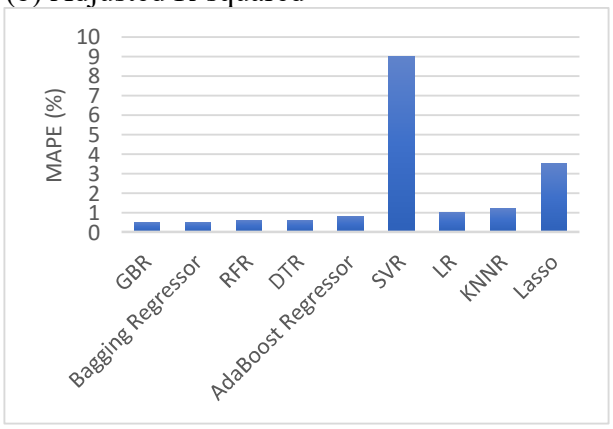
(a) MAE



(b) Adjusted R-squared



(c) RMSE



(d) MAPE

Figure 13. Performance comparison of ML models for Cooling Load estimation using MAE, Adjusted R^2, RMSE, and MAPE.

Leave-One-Out Cross-Validation (LOO-CV), a special case of cross-validation, was employed to evaluate the performance of the machine learning models. In this approach, a single data sample is set aside for testing while the model is trained on all remaining samples, and the process is repeated until each sample has been used once as a test instance. This method is particularly suitable for the present dataset, where maximizing the use of all available data is essential. By ensuring that every observation contributes to both training and testing, LOO-CV offers a comprehensive and unbiased assessment of model performance. The evaluation results for the Gradient Boosting Regressor under LOO-CV are illustrated in Figure 14.

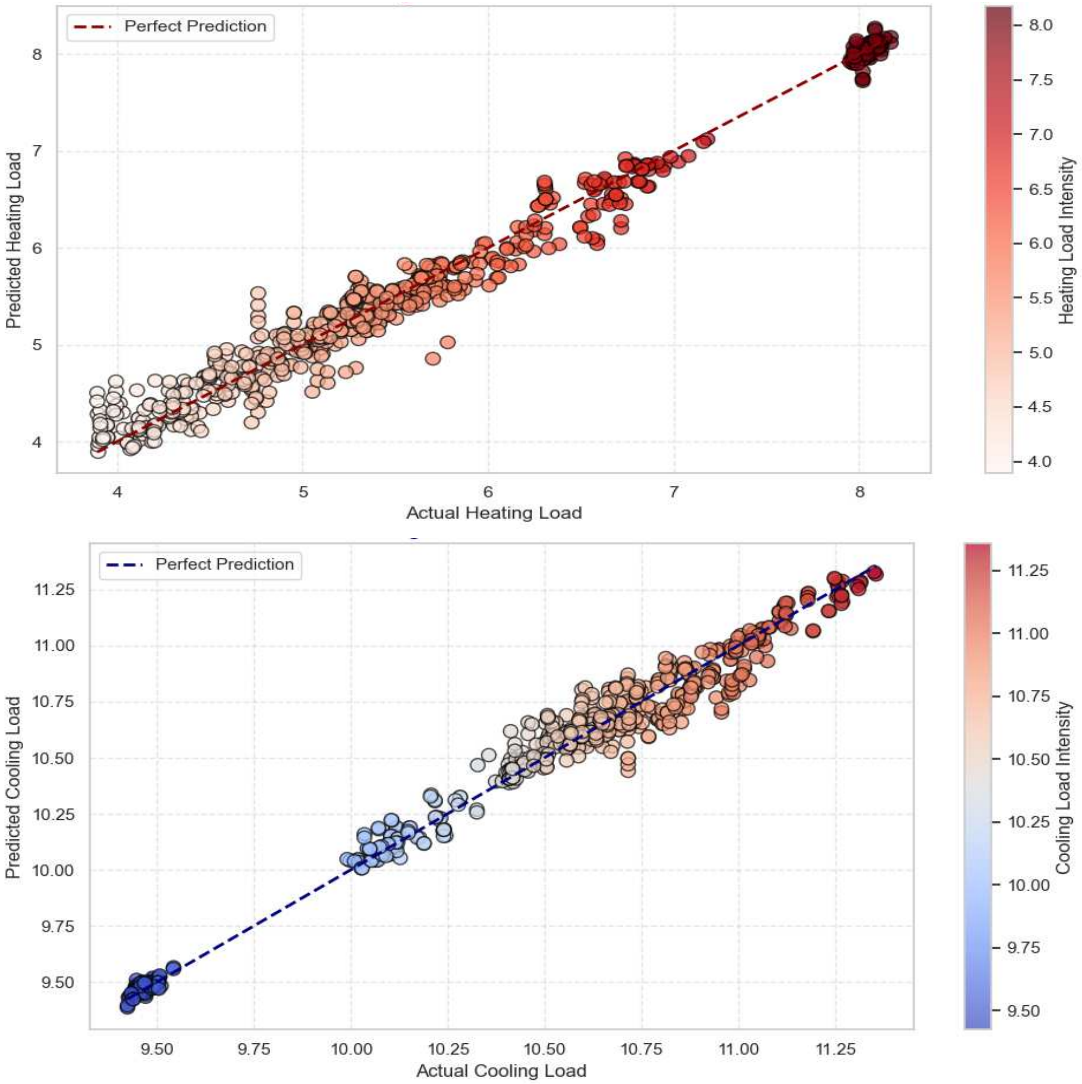


Figure 14. Model Performance using Leave-One-Out Cross-Validation (Actual vs Predicted Values)

Feature selection techniques were systematically applied during model training rather than removing features manually, to reduce redundancy and maintain objectivity. An adaptive feature ranking approach was adopted, combining flexibility and accuracy in identifying the most relevant predictors. Specifically, feature importance from tree-based models and Recursive Feature Elimination (RFE) were employed to establish a scalable strategy applicable across different learning algorithms. This ranking process not only identifies the features most influential for heating and cooling load prediction but also enhances model robustness and interpretability, supporting data-driven design decisions. Figure 15 illustrates the feature ranking obtained using the Gradient Boosting Regressor, where the window-to-wall ratio (X7) and glazing area distribution (X8) emerged as the most influential variables for both heating and cooling demands.

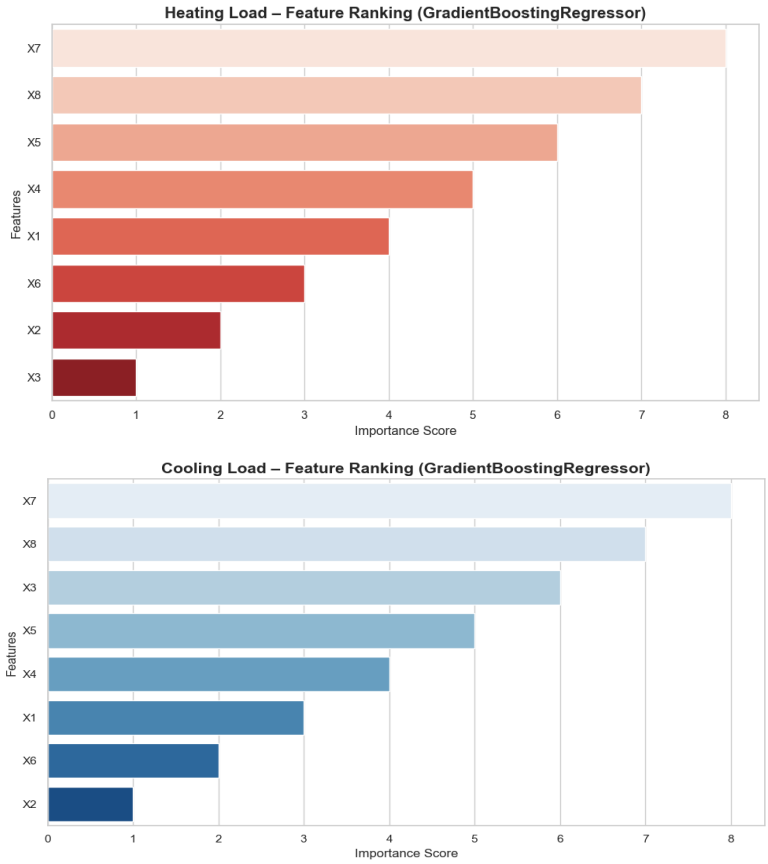


Figure 15. Feature ranking of Gradient Boosting Regressor for Heating and Cooling Loads.

In this paper, the prediction performance of nine ML techniques—Gradient Boosting Regressor, Random Forest Regressor, Decision Tree Regressor, Bagging Regressor, AdaBoost Regressor, K Neighbors Regressor, SVR, LR, and Lasso—is compared for predicting both HL and CL. The study's results highlight an ML model that empowers structure designers to predict energy loads during the initial design stage. Overall, the Gradient Boosting Regressor yielded superior results for energy prediction. However, other efficient predictive models were also identified in this study, including Random Forest Regressor, Decision Tree Regressor, Bagging Regressor, and AdaBoost Regressor.

6. Conclusion and Future Work

Energy efficiency has become a central concern in modern engineering due to its substantial economic and environmental implications. Accurate prediction of heating and cooling loads plays a crucial role in minimizing energy consumption and mitigating risks related to climate change. This study demonstrated that machine learning (ML) algorithms can serve as effective tools for forecasting thermal loads by analyzing key building characteristics such as geometry, insulation, and glazing configurations. The results confirmed that ensemble-based approaches, particularly Gradient Boosting and Bagging Regressors, significantly outperform traditional regression techniques, achieving adjusted R^2 values exceeding 0.97 for both heating and cooling loads. These findings underscore the reliability of ML-driven prediction models and their potential contribution to energy-efficient building design. Beyond predictive accuracy, feature importance analysis revealed that window-to-wall ratio and glazing distribution are dominant factors affecting thermal

performance. Nevertheless, our contribution is the contextualization of these insights inside a replicable simulation framework specifically designed for Egyptian residential buildings. This insight provides practical guidance for architects and engineers seeking to integrate energy-saving considerations into the early stages of building design. Future research could extend this work in several directions. First, applying the proposed framework to larger and more diverse real-world datasets would enhance its generalizability. Second, incorporating temporal and climatic data could improve model adaptability to varying environmental conditions. Third, hybrid approaches that combine ML with physics-based simulations may yield more robust predictions. Finally, exploring deployment in decision-support systems can bridge the gap between academic research and practical applications in sustainable building design.

Funding sources

This research did not receive any specific grant from funding agencies in the public, commercial, or not-for-profit sectors.

Competing Interests

The authors declare that they have no known competing financial or non-financial interests that could have appeared to influence the work reported in this paper.

Data Availability

The dataset supporting the findings of this study is available in the repository titled Predicting Thermal Energy Demands in Residential Buildings [65]. The repository record is publicly accessible; however, the dataset files are currently restricted and can only be accessed by users specified in the permissions. The data is available from the corresponding author upon reasonable request and will be made publicly available upon publication.

Ethics declaration:

Not applicable.

Reference

[1] H. Zlotnik, "World urbanization: trends and prospects," in *New forms of urbanization*, Routledge, 2017, pp. 43–64.

[2] Q. Wang, J. Lin, K. Zhou, J. Fan, and M.-P. Kwan, "Does urbanization lead to less residential energy consumption? A comparative study of 136 countries," *Energy*, vol. 202, p. 117765, 2020, doi: https://doi.org/10.1016/j.energy.2020.117765.

[3] I. Citaristi, "International energy agency—iea," in *The Europa directory of international organizations 2022*, Routledge, 2022, pp. 701–702.

[4] F. S. Hafez *et al.*, "Energy Efficiency in Sustainable Buildings: A Systematic Review with Taxonomy, Challenges, Motivations, Methodological Aspects, Recommendations, and Pathways for Future Research," *Energy Strategy Reviews*, vol. 45, p. 101013, 2023, doi: https://doi.org/10.1016/j.esr.2022.101013.

[5] U. Ali, M. H. Shamsi, C. Hoare, E. Mangina, and J. O'Donnell, "Review of urban building energy modeling (UBEM) approaches, methods and tools using qualitative and quantitative analysis," *Energy and Buildings*, vol. 246, p. 111073, 2021, doi: https://doi.org/10.1016/j.enbuild.2021.111073.

[6] F. Ascione, N. Bianco, R. F. D. Masi, F. de' Rossi, and G. P. Vanoli, "Energy refurbishment of existing buildings through the use of phase change materials: Energy savings and indoor comfort in the cooling season," *Applied Energy*, vol. 113, pp. 990–1007, 2014, doi: https://doi.org/10.1016/j.apenergy.2013.08.045.
[7] M. Esrafilian-Najafabadi and F. Haghighat, "Occupancy-based HVAC control using deep learning algorithms for estimating online preconditioning time in residential buildings," *Energy and Buildings*, vol. 252, p. 111377, 2021, doi: https://doi.org/10.1016/j.enbuild.2021.111377.
[8] J. Lin *et al.*, "Predictive analytics for building power demand: Day-ahead forecasting and anomaly prediction," *Energy and Buildings*, vol. 255, p. 111670, 2022, doi: https://doi.org/10.1016/j.enbuild.2021.111670.
[9] M.-M. Fernandez-Antolin, J. M. del Río, and R.-A. Gonzalez-Lezcano, "Building performance simulation tools as part of architectural design: breaking the gap through software simulation," *International Journal of Technology and Design Education*, vol. 32, no. 2, pp. 1227–1245, Apr. 2022, doi: 10.1007/s10798-020-09641-7.
[10] G. Salvalai, Y. Zhu, and M. M. Sesana, "From building energy modeling to urban building energy modeling: A review of recent research trend and simulation tools," *Energy and Buildings*, vol. 319, p. 114500, 2024, doi: https://doi.org/10.1016/j.enbuild.2024.114500.
[11] R. Mahmoud, J. M. Kamara, and N. Burford, "Opportunities and Limitations of Building Energy Performance Simulation Tools in the Early Stages of Building Design in the UK," *Sustainability*, vol. 12, no. 22, 2020, doi: 10.3390/su12229702.
[12] K. Reindl and J. Palm, "Energy efficiency in the building sector: a combined middle-out and practice theory approach," *IJSEPM*, vol. 28, pp. 3–16, Apr. 2020, doi: 10.5278/ijsepm.3426.
[13] B. Bass, J. New, N. Clinton, M. Adams, B. Copeland, and C. Amoo, "How close are urban scale building simulations to measured data? Examining bias derived from building metadata in urban building energy modeling," *Applied Energy*, vol. 327, p. 120049, Dec. 2022, doi: 10.1016/j.apenergy.2022.120049.
[14] J. Yu, W.-S. Chang, and Y. Dong, "Building Energy Prediction Models and Related Uncertainties: A Review," *Buildings*, vol. 12, no. 8, 2022, doi: 10.3390/buildings12081284.
[15] D. Senthil Kumar, D. George Washington, A. K. Reshmy, and M. Noorunnisha, "Energy Performance Prediction of Residential Buildings Using Nonlinear Machine Learning Technique," in *International Conference on Innovative Computing and Communications*, A. Khanna, D. Gupta, S. Bhattacharyya, A. E. Hassanien, S. Anand, and A. Jaiswal, Eds., Singapore: Springer Singapore, 2022, pp. 81–93.
[16] F. Bünning *et al.*, "Physics-informed linear regression is competitive with two Machine Learning methods in residential building MPC," *Applied Energy*, vol. 310, p. 118491, 2022, doi: https://doi.org/10.1016/j.apenergy.2021.118491.
[17] R. Morcillo-Jimenez, J. Mesa, J. Gómez-Romero, M. A. Vila, and M. J. Martin-Bautista, "Deep learning for prediction of energy consumption: an applied use case in an office building," *Applied Intelligence*, vol. 54, no. 7, pp. 5813–5825, Apr. 2024, doi: 10.1007/s10489-024-05451-9.
[18] C. J. Hopfe and J. L. M. Hensen, "Uncertainty analysis in building performance simulation for design support," *Energy and Buildings*, vol. 43, no. 10, pp. 2798–2805, 2011, doi: https://doi.org/10.1016/j.enbuild.2011.06.034.
[19] A. Tsanas and A. Xifara, "Energy Efficiency." 2012.
[20] A. Tsanas and A. Xifara, "Accurate quantitative estimation of energy performance of residential buildings using statistical machine learning tools," *Energy and Buildings*, vol. 49, pp. 560–567, 2012, doi: https://doi.org/10.1016/j.enbuild.2012.03.003.
[21] M. Al-Rakhami, A. Gumaei, A. Alsanad, A. Alamri, and M. M. Hassan, "An Ensemble Learning Approach for Accurate Energy Load Prediction in Residential Buildings," *IEEE Access*, vol. 7, pp. 48328–48338, 2019, doi: 10.1109/ACCESS.2019.2909470.
[22] B. Mehdizadeh Khorrami, A. Soleimani, A. Pinnarelli, G. Brusco, and P. Vizza, "Forecasting heating and cooling loads in residential buildings using machine learning: a comparative study of techniques

and influential indicators," *Asian Journal of Civil Engineering*, vol. 25, no. 2, pp. 1163–1177, Feb. 2024, doi: 10.1007/s42107-023-00834-8.
[23] M. Mohebbi and S. Afzal, "Enhancing residential heating load prediction with advanced machine learning and optimization techniques," *Journal of Building Engineering*, vol. 95, p. 110199, 2024, doi: https://doi.org/10.1016/j.jobe.2024.110199.
[24] B. Sadaghat, S. Afzal, and A. J. Khiavi, "Residential building energy consumption estimation: A novel ensemble and hybrid machine learning approach," *Expert Systems with Applications*, vol. 251, p. 123934, 2024, doi: https://doi.org/10.1016/j.eswa.2024.123934.
[25] S. Afzal, B. M. Ziapour, A. Shokri, H. Shakibi, and B. Sobhani, "Building energy consumption prediction using multilayer perceptron neural network-assisted models; comparison of different optimization algorithms," *Energy*, vol. 282, p. 128446, 2023, doi: https://doi.org/10.1016/j.energy.2023.128446.
[26] Z. Dong, J. Liu, B. Liu, K. Li, and X. Li, "Hourly energy consumption prediction of an office building based on ensemble learning and energy consumption pattern classification," *Energy and Buildings*, vol. 241, p. 110929, 2021, doi: https://doi.org/10.1016/j.enbuild.2021.110929.
[27] C. Miller *et al.*, "The Building Data Genome Project 2, energy meter data from the ASHRAE Great Energy Predictor III competition," *Scientific Data*, vol. 7, no. 1, p. 368, Oct. 2020, doi: 10.1038/s41597-020-00712-x.
[28] R. Olu-Ajayi, H. Alaka, I. Sulaimon, F. Sunmola, and S. Ajayi, "Building energy consumption prediction for residential buildings using deep learning and other machine learning techniques," *Journal of Building Engineering*, vol. 45, p. 103406, 2022, doi: https://doi.org/10.1016/j.jobe.2021.103406.
[29] F. Dinmohammadi, Y. Han, and M. Shafiee, "Predicting Energy Consumption in Residential Buildings Using Advanced Machine Learning Algorithms," *Energies*, vol. 16, no. 9, 2023, doi: 10.3390/en16093748.
[30] S. Kumar, S. K. Pal, and R. P. Singh, "A novel method based on extreme learning machine to predict heating and cooling load through design and structural attributes," *Energy and Buildings*, vol. 176, pp. 275–286, 2018, doi: https://doi.org/10.1016/j.enbuild.2018.06.056.
[31] R. Wang, S. Lu, and W. Feng, "A novel improved model for building energy consumption prediction based on model integration," *Applied Energy*, vol. 262, p. 114561, 2020, doi: https://doi.org/10.1016/j.apenergy.2020.114561.
[32] A. Sepehr and S. Dehghani, "Energy and Environmental Study of HVAC Systems for a High-rise Building using Simulation by DesignBuilder and RetScreen," in *2024 9th International Conference on Technology and Energy Management (ICTEM)*, 2024, pp. 1–5. doi: 10.1109/ICTEM60690.2024.10631964.
[33] L. Waas and Enjellina, "Review of BIM-Based Software in Architectural Design: Graphisoft Archicad VS Autodesk Revit," *Journal of Artificial Intelligence in Architecture*, vol. 1, no. 2, pp. 14–22, Aug. 2022, doi: 10.24002/jarina.v1i2.6016.
[34] T. A. Alghamdi and N. Javaid, "A Survey of Preprocessing Methods Used for Analysis of Big Data Originated From Smart Grids," *IEEE Access*, vol. 10, pp. 29149–29171, 2022, doi: 10.1109/ACCESS.2022.3157941.
[35] D. W. Scott, "Histogram," *WIREs Computational Statistics*, vol. 2, no. 1, pp. 44–48, 2010, doi: https://doi.org/10.1002/wics.59.
[36] A. Ghosh, M. Nashaat, J. Miller, and S. Quader, "Context-Based Evaluation of Dimensionality Reduction Algorithms—Experiments and Statistical Significance Analysis," *ACM Transactions on Knowledge Discovery from Data*, vol. 15, no. 2, Jan. 2021, doi: 10.1145/3428077.
[37] Z. Gu, "Complex heatmap visualization," *iMeta*, vol. 1, no. 3, p. e43, 2022, doi: https://doi.org/10.1002/imt2.43.
[38] A. Ghosh, M. Nashaat, J. Miller, and S. Quader, "VisExPreS: A Visual Interactive Toolkit for User-Driven Evaluations of Embeddings," *IEEE Transactions on Visualization and Computer Graphics*, vol. 28, no. 7, pp. 2791–2807, Jul. 2022, doi: 10.1109/TVCG.2020.3039106.

[39] J. I. Daoud, "Multicollinearity and Regression Analysis," *Journal of Physics: Conference Series*, vol. 949, no. 1, p. 012009, Dec. 2017, doi: 10.1088/1742-6596/949/1/012009.
[40] E. Esenogho, I. D. Mienye, T. G. Swart, K. Aruleba, and G. Obaido, "A Neural Network Ensemble With Feature Engineering for Improved Credit Card Fraud Detection," *IEEE Access*, vol. 10, pp. 16400–16407, 2022, doi: 10.1109/ACCESS.2022.3148298.
[41] H. Jeon and S. Oh, "Hybrid-Recursive Feature Elimination for Efficient Feature Selection," *Applied Sciences*, vol. 10, no. 9, 2020, doi: 10.3390/app10093211.
[42] Z. Zhou and G. Hooker, "Unbiased Measurement of Feature Importance in Tree-Based Methods," *ACM Trans. Knowl. Discov. Data*, vol. 15, no. 2, Jan. 2021, doi: 10.1145/3429445.
[43] P. Rodríguez, M. A. Bautista, J. Gonzàlez, and S. Escalera, "Beyond one-hot encoding: Lower dimensional target embedding," *Image and Vision Computing*, vol. 75, pp. 21–31, 2018, doi: https://doi.org/10.1016/j.imavis.2018.04.004.
[44] D. Bollegala, "Dynamic feature scaling for online learning of binary classifiers," *Knowledge-Based Systems*, vol. 129, pp. 97–105, 2017, doi: https://doi.org/10.1016/j.knosys.2017.05.010.
[45] M. M. Ahsan, M. A. P. Mahmud, P. K. Saha, K. D. Gupta, and Z. Siddique, "Effect of Data Scaling Methods on Machine Learning Algorithms and Model Performance," *Technologies*, vol. 9, no. 3, 2021, doi: 10.3390/technologies9030052.
[46] H. Wu and D. Levinson, "The ensemble approach to forecasting: A review and synthesis," *Transportation Research Part C: Emerging Technologies*, vol. 132, p. 103357, 2021, doi: https://doi.org/10.1016/j.trc.2021.103357.
[47] Y. Wang, Y. Hu, and K.-K. P. and, "Non-parametric modelling and simulation of spatiotemporally varying geo-data," *Georisk: Assessment and Management of Risk for Engineered Systems and Geohazards*, vol. 16, no. 1, pp. 77–97, 2022, doi: 10.1080/17499518.2021.1971258.
[48] D. Valero-Carreras, J. Alcaraz, and M. Landete, "Comparing two SVM models through different metrics based on the confusion matrix," *Computers & Operations Research*, vol. 152, p. 106131, 2023, doi: https://doi.org/10.1016/j.cor.2022.106131.
[49] D. A. Otchere, T. O. A. Ganat, J. O. Ojero, B. N. Tackie-Otoo, and M. Y. Taki, "Application of gradient boosting regression model for the evaluation of feature selection techniques in improving reservoir characterisation predictions," *Journal of Petroleum Science and Engineering*, vol. 208, p. 109244, 2022, doi: https://doi.org/10.1016/j.petrol.2021.109244.
[50] N. Pachauri and C. W. Ahn, "Regression tree ensemble learning-based prediction of the heating and cooling loads of residential buildings," *Building Simulation*, vol. 15, no. 11, pp. 2003–2017, Nov. 2022, doi: 10.1007/s12273-022-0908-x.
[51] F. Aslam, R. Alyousef, H. H. Awan, and M. F. Javed, "Forecasting the self-healing capacity of engineered cementitious composites using bagging regressor and stacking regressor," *Structures*, vol. 54, pp. 1717–1728, 2023, doi: https://doi.org/10.1016/j.istruc.2023.05.140.
[52] G. Shanmugasundar, M. Vanitha, R. Čep, V. Kumar, K. Kalita, and M. Ramachandran, "A Comparative Study of Linear, Random Forest and AdaBoost Regressions for Modeling Non-Traditional Machining," *Processes*, vol. 9, no. 11, 2021, doi: 10.3390/pr9112015.
[53] M. Nashaat, A. Ghosh, J. Miller, and S. Quader, "Asterisk: Generating Large Training Datasets with Automatic Active Supervision," *ACM/IMS Transactions on Data Science*, vol. 1, no. 2, May 2020, doi: 10.1145/3385188.
[54] H. Luo, F. Cheng, H. Yu, and Y. Yi, "SDTR: Soft Decision Tree Regressor for Tabular Data," *IEEE Access*, vol. 9, pp. 55999–56011, 2021, doi: 10.1109/ACCESS.2021.3070575.
[55] Y. Nader, L. Sixt, and T. Landgraf, "DNNR: Differential Nearest Neighbors Regression," in *Proceedings of the 39th International Conference on Machine Learning*, K. Chaudhuri, S. Jegelka, L. Song, C. Szepesvari, G. Niu, and S. Sabato, Eds., in Proceedings of Machine Learning Research, vol. 162. PMLR, Jul. 2022, pp. 16296–16317. [Online]. Available: https://proceedings.mlr.press/v162/nader22a.html
[56] A. M. Ahmed, A. Rizaner, and A. H. Ulusoy, "A Decision Tree Algorithm Combined with Linear Regression for Data Classification," in *2018 International Conference on Computer, Control,*

Electrical, and Electronics Engineering (ICCCEEE), 2018, pp. 1–5. doi: 10.1109/ICCCEEE.2018.8515759.

[57] M. Yazdi, N. A. Golilarz, A. Nedjati, and K. A. Adesina, "An improved lasso regression model for evaluating the efficiency of intervention actions in a system reliability analysis," *Neural Computing and Applications*, vol. 33, no. 13, pp. 7913–7928, Jul. 2021, doi: 10.1007/s00521-020-05537-8.

[58] X. Deng *et al.*, "Bagging–XGBoost algorithm based extreme weather identification and short-term load forecasting model," *Energy Reports*, vol. 8, pp. 8661–8674, 2022, doi: https://doi.org/10.1016/j.egyr.2022.06.072.

[59] T. O. Hodson, "Root-mean-square error (RMSE) or mean absolute error (MAE): when to use them or not," *Geoscientific Model Development*, vol. 15, no. 14, pp. 5481–5487, 2022, doi: 10.5194/gmd-15-5481-2022.

[60] Solimun and A. A. R. Fernandes, "Ensemble Bagging Discriminant and Logistic Regression in Classification Analysis," *New Mathematics and Natural Computation*, vol. 21, no. 01, pp. 91–111, 2025, doi: 10.1142/S1793005725500061.

[61] M. Nashaat, A. Ghosh, J. Miller, S. Quader, and C. Marston, "M-Lean: An end-to-end development framework for predictive models in B2B scenarios," *Information and Software Technology*, vol. 113, pp. 131–145, Sep. 2019, doi: 10.1016/j.infsof.2019.05.009.

[62] C. G. Raju, V. Amudha, and S. G, "Comparison of Linear Regression and Logistic Regression Algorithms for Ground Water Level Detection with Improved Accuracy," in *2023 Eighth International Conference on Science Technology Engineering and Mathematics (ICONSTEM)*, 2023, pp. 1–6. doi: 10.1109/ICONSTEM56934.2023.10142495.

[63] M.-K. Tran *et al.*, "Python-based scikit-learn machine learning models for thermal and electrical performance prediction of high-capacity lithium-ion battery," *International Journal of Energy Research*, vol. 46, no. 2, pp. 786–794, 2022, doi: https://doi.org/10.1002/er.7202.

[64] D. S. K. Karunasingha, "Root mean square error or mean absolute error? Use their ratio as well," *Information Sciences*, vol. 585, pp. 609–629, 2022, doi: https://doi.org/10.1016/j.ins.2021.11.036.

[65] Nashaat, B., Nashaat, M., & El-Shekheby, S. (2025). Predicting Thermal Energy Demands in Residential Buildings [Dataset]. Zenodo. https://doi.org/10.5281/zenodo.16954171