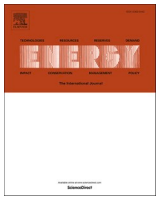
Energy 308 (2024) 132979

Contents lists available at ScienceDirect

Energy

journal homepage: www.elsevier.com/locate/energy



Evaluating missing data handling methods for developing building energy benchmarking models

Kyungjae Lee [a], Hyunwoo Lim [b,*], Jeongyun Hwang [a], Doyeon Lee [a]

[a] Department of Architecture, Graduate School of Konkuk University, Seoul, South Korea
[b] Department of Architecture, College of Architecture, Konkuk University, Seoul, South Korea

ARTICLE INFO

Handling editor: X Zhao

Keywords:
Building energy benchmarking model
Building energy performance
Building energy data
Missing value imputation
Machine learning

ABSTRACT

This study explored methods for handling missing data in the development of machine learning-based energy benchmarking models, assessing their training time, performance, and variance. Unlike the common assumption of missing completely at random, this study adopted a missing at random (MAR) perspective, which is more appropriate for building data. We compared the inherent missing data handling method of extreme gradient boosting (XGBoost) with the Median, k-nearest neighbors (KNN), and classification and regression trees (CART) methods, alongside Shapley additive explanation (SHAP) method for model interpretability. The findings indicate that, despite its computational demands, the CART method most accurately mirrors the original data distribution, thereby enhancing model performance and stability. The KNN method is effective, while the XGBoost method is viable under computational time constraints. This work highlights the importance of reliable test data for performing accurate evaluations of imputation methods. These results offer guidelines for the selection of imputation methods in model development, contributing to the improved accuracy of energy benchmarking models. The MAR-based approach for missing data analysis holds promise for future research on building energy data, providing crucial insights for accurate energy benchmark model performance assessments.

1. Introduction

1.1. Research background and motivation

According to a 2022 International Energy Agency (IEA) report, the building sector accounts for 27 % of energy consumption and 33 % of greenhouse gas emissions, highlighting its significance [1]. Simultaneously, buildings offer significant potential for sustainable carbon reduction among all economic sectors [2]. Consequently, many countries have begun enacting laws and retrofitting older buildings to enhance energy efficiency [3,4]. Additionally, research on building energy benchmarking models is being actively pursued to assess energy performance and identify inefficiencies. A notable example is the ENERGY STAR Portfolio Manager, created by the U.S. Department of Energy, which, according to the Protecting Agency Report, contributed to 7 % energy savings in 35,000 buildings over four years [5].

Energy benchmarking can be divided into two approaches. The first approach involves energy simulation based on physical building models, such as EnergyPlus, DOE-2, and TRNSYS. This approach uses energy simulation software to compare simulated values with actual energy usage [6]. One advantage of the physical model is that it allows for a detailed analysis of building energy consumption based on specific inputs, such as building structure, materials, HVAC systems, lighting, and other operational information. This enables the identification of factors that substantially impact energy consumption and the application of various energy conservation measures. However, data collection is challenging, and specialized knowledge and time are required to create simulation models for each building, posing challenges for citywide large-scale energy benchmarking.

An alternative approach is the data-driven method, which outperforms large-scale energy benchmarks. Unlike physical models, this method leverages statistical models or machine learning techniques to analyze data patterns for energy benchmarking without the need for physical models [7]. This approach can swiftly analyze previously collected data and be broadly applied across various building types and energy usage patterns. The ENERGY STAR benchmark adopts a data-driven approach, utilizing national databases to construct building energy consumption models using multiple linear regression (MLR). This method quantifies the effects of key variables related to energy use and generates the estimated energy use intensity (EUI) [8]. However, these linear models are unable to accurately capture the nonlinear

* Corresponding author. 120 Neungdong-ro, Gwangjin-gu, Seoul, 05029, South Korea.
E-mail address: hyunwoolim@konkuk.ac.kr (H. Lim).

https://doi.org/10.1016/j.energy.2024.132979
Received 22 March 2024; Received in revised form 5 July 2024; Accepted 24 August 2024
Available online 30 August 2024

Glossary

Acronyms, abbreviations

CART	Classification and regression trees
EDA	Exploratory data analysis
EMD	Earth mover's distance
FAR	Floor area ratio
GBDT	Gradient boosting decision trees
KNN	K-nearest neighbor imputation
MAR	Missing at random
MCAR	Missing completely at random
Median	Median imputation
MI	Multiple imputation
MLR	Multiple linear regression
MNAR	Missing not at random
R^2	R-squared
SHAP	Shapley additive explanations
XAI	Explainable artificial intelligence
\XGBoost	Extreme gradient boosting algorithm
XGBoost	The inherent missing data handling method of the XGBoost algorithm
γ	Pearson correlation coefficient

relationships among the elements within buildings related to energy consumption, leading to lower-performing energy models [9]. To address the limitations of linear models, research is being conducted to improve model accuracy using nonlinear models, such as support vector machines (SVM), artificial neural networks (ANN), and extreme gradient boosting (XGBoost) [10–13]. Among recent algorithms, XGBoost, based on gradient boosting decision trees (GBDT), has gained considerable attention [2]. This method is faster and more accurate than conventional machine learning algorithms. When combined with the explainable artificial intelligence (XAI) method, specifically the Shapley additive explanation (SHAP), it can identify variable importance and the impact of variable values on the model [9,14]. This advantage has led to its widespread use in various building energy benchmarking studies.

The success of data-driven approaches is heavily influenced not only by the selection of an appropriate machine learning algorithm but also by data quality [2]. Data quality includes both quantitative and qualitative aspects, such as outliers, missing values, and feature selection. In this study, we focused on missingness, which is a key qualitative component of data quality. Improperly handled missingness can lead to biased results, which, in turn, can lead to incorrect model predictions and undermine study reliability [15]. Therefore, this study aims to build a more accurate and reliable energy benchmarking model by focusing on the impact of missing data handling methods on model performance and to propose guidelines for missing data handling for building energy benchmarking models.

1.2. Literature review of methodologies for handling missing values

Table 1 summarizes research papers related to missing data handling over the last seven years. In the "missing scenario" column, "-" indicates cases where the research did not explicitly mention the missing data scenario. According to the table, research in the fields of energy or environment primarily focuses on handling missing data in time-series data, such as energy consumption data [16–21]. Missing data in a time series significantly impacts model performance owing to disrupted data continuity, making it a critical research topic in the energy sector. However, most studies provide unclear definitions of missing data. This is because missingness in time-series data is often handled using model-based approaches that focus on identifying patterns in missing data rather than the causes of missingness [15]. Identifying and defining the causes of missing data beforehand plays a crucial role in understanding the dataset and developing future data collection strategies. Thus, it is advisable to conduct this process in the early stages of research. Among the research on missing data handling in the energy sector, Kim et al. (2019) [22] addressed missing solar power generation

Table 1
Summary of previous studies on missing data handling.

Author	Problem	Missing scenario	Handling method	Domain of application	Year	Reference
Ma et al.	Time series	MCAR & continuous missing	LSTM	Energy	2020	[16]
Li et al.	Time series	–	BPNN	Energy	2020	[17]
Liu and Zhang	Time series	–	SAE-CD	Energy	2021	[18]
Hussain et al.	Time series	–	CNN-LSTM	Energy	2022	[19]
Jung et al.	Time series	–	SENet	Energy	2020	[20]
Kim et al.	Prediction	MCAR	Median, LI, KNN, MICE	Energy	2019	[22]
Kim et al.	Prediction	MCAR	MLR, RF, SVM, MLP	Energy	2023	[23]
Wijesekara and Liyanage	Time series	MCAR	Mean, LI, SMA, EMA, Kalman Smoothing on Time series model, Kalman smoothing on ARIMA	Environmental	2020	[21]
Fauzan and Mrufi	Prediction	MAR	XGBoost, Mean, Median	Life sciences and health	2018	[24]
Rusdah and Murfi	Classification	MCAR	XGBoost, Mean, KNN	Life sciences and health	2020	[25]
Bertsmias et al.	Prediction	MCAR	Mean, PMM, Bayesian PCA, KNN, iterative KNN, opt.impute	Life sciences and others	2018	[26]
Jadhav et al.	Prediction	MCAR	Mean, Median, KNN, Mean, Median, KNN, PMM, norm, norm.nob, Random replacement	Miscellaneous	2019	[27]
Emmanuel et al.	Post imputation comparison	MCAR	KNN, RF	Miscellaneous	2021	[28]
Sanjar et al.	Prediction	–	Mean, KNN, KNN-MCF	Social sciences	2020	[29]
Akande et al.	Classification	MCAR	MI-GLM, MI-CART, MI-DPM	Social sciences	2017	[30]
Poulos and Valle	Classification	MCAR & MNAR	Random replacement, KNN, Logistic regression, RF, SVMs, Median	Social sciences	2018	[31]

* Autoregressive integrated moving average (ARIMA), bi-directional imputation and transfer learning (BIT), back propagation neural network (BPNN), classification and regressions trees (CART), coordinate descendant (CD), convolutional neural network (CNN), Dirichlet process mixture (DPMPM), exponentially weighted moving average (EMA), generalized linear models (GLM), k-nearest neighbors (KNN), long short-term memory (LSTM), multilayer perceptron neural network (MLP), Bayesian linear regression (norm), linear regression, non-Bayesian (norm.nob), optimized imputation framework (opt.impute), predictive-mean matching (PMM), sparse autoencoder (SAE), softmax ensemble network (SENet), and simple moving average (SMA).

data using the k-nearest neighbors (KNN) method, and Kim et al. (2023) [23] explored optimal missing data handling methods in residential building monitoring data.

Research on handling missing input data has been predominantly conducted outside the building energy sector. For example, Bertsimas et al. proposed a novel missing data handling method called the optimized imputation framework [26]. This approach outperformed conventional imputation methods (Mean, KNN, iterative KNN, Bayesian PCA, and predictive-mean matching (PMM)) in most datasets, highlighting its effectiveness in managing missing data more efficiently. Emmanuel et al. [28] evaluated the performance of KNN and random forest in imputing missing data under missing completely at random (MCAR) conditions. Sanjar et al. [29] conducted research on imputing missing data for geographical data-based price prediction problems using the KNN algorithm based on the most correlated features (KNN–MCF), and Poulos and Valle [31] found that KNN exhibited the best performance in classifying problems in supervised learning models under MCAR conditions.

In a study by Jadhav et al. [27], the performance of seven missing data imputation methods (Mean, Median, KNN, PMM, Bayesian linear regression (norm), non-Bayesian linear regression (norm.nob), and random replacement) was compared for continuous variables under MCAR conditions using normalized root mean squared error. The KNN imputation method outperformed the other methods. Akande et al. [30] investigated the performance of missing data handling methods using multiple imputation generalized linear models (MI-GLM), multiple imputation classification and regression tree (MI-CART), and multiple imputation Dirichlet process mixtures (MI-DPM) for categorical variables. The results indicate that MI-CART and MI-DPM are superior at imputing missing data for categorical variables. In addition, this study highlights the flexibility of the MI-CART method across various missing data scenarios [15].

Recently, in the field of machine learning, attention has shifted from conventional missing data imputation methods to utilizing the inherent algorithm of XGBoost for missing data handling without imputation. In a study by Fauzan and Murfi [24], the XGBoost missing data handling method showed superior performance compared to the Mean and Median methods. Similarly, Rusdah and Murfi [25] confirmed that for classification problems, the XGBoost method outperformed the Mean and KNN methods in terms of model performance. These findings are noteworthy because the XGBoost method for missing data handling does not require prior processing by users. However, there is still a lack of interest in utilizing the inherent XGBoost method to handle missing data in the building energy sector.

Based on previous studies, this study analyzed four missing data handling methods for constructing building energy benchmarking models. These methods are the basic Median; the KNN, which showed superior performance for continuous variables; the CART method, which excelled in classification problems; and the inherent XGBoost missing data handling method. These methods were compared in terms of computation time, model performance, and model deviation to identify the optimal missing data handling approach for building energy benchmarking models.

1.3. Novelty and contribution of this study

This work highlights the importance of handling missing data in building energy benchmarking models. Unlike previous research, which primarily focused on missing target value data (i.e., energy consumption) [16–21], this study concentrates on missing input data. Most existing studies that assign missing data using the MCAR method introduce missing at random (MAR), a missingness assignment method that considers the characteristics of building energy data. This approach allows for a more accurate reflection of real-world missing data scenarios and a better understanding of model performance changes owing to different missing data handling methods. Additionally, it makes statistical sense, as most missing data handling methods are based on the MAR assumption [15].

After introducing MAR missing data, this study meticulously analyzed the effects of various missing data handling methods on processing time, model performance, and deviation, offering practical guidelines for optimal missing data handling. This is a key contribution of this research, as these guidelines assist in systematically selecting data preprocessing techniques for building energy models, potentially enhancing model performance. In addition, enhancing model interpretability through SHAP analysis aids in identifying the key variables that influence model predictions and provides valuable insights to stakeholders [2].

2. Methodology

This study investigated the impact of different missing data handling methods on the construction of building energy benchmarking models. To achieve this, we examined the differences between each missing data handling method by: 1) analyzing the computation time and the difference in distribution with the original data, and 2) evaluating the performance differences of the energy benchmarking models based on the missing data handling method.

The research stages, shown in Fig. 1, are as follows. In the data collection and preparation stage, data were collected and cleaned to create an initial dataset without missing values (Case 0) (Section 2.1). The data-split stage divided the Case 0 dataset into training and test sets, where the training set was used to generate and handle missing data, and for model training. The test set was used to assess model performance based on the missing data handling method (Section 2.2). Assigning the missing values involved allocating missing data at rates of 20 %, 40 %, and 60 % to Case 0 based on the variable type, differentiating it into: Case 1, Case 1-1, …Case 3-3 (Section 2.3). In the missing data handling stage, missing data handling methods such as XGBoost, Median, KNN, and CART were applied to each case (Section 2.4), followed by an evaluation and comparison stage to identify the computation time and distribution differences from the original data for each missing data handling method (Section 2.5). The model development stage built a building energy benchmarking model based on the data treated for missing values in each case and evaluated the model performance variability depending on the missing data handling method through a 10-fold cross-validation (Section 2.6). Finally, in the evaluation and interpretation stage, the test set, which was divided in the data-split stage, was used to compare the performance of models based on the missing data handling method and interpret the model through SHAP analysis (Section 2.7).

2.1. Data collection and preparation

The data used in this study were obtained from a data-centric check-up database created by the Korea Institute of Civil Engineering and Building Technology, a national research institute in Korea, based on information from the Building Register of South Korea [32]. These data-centric check-up data, which are not publicly available, include basic building information and the total energy usage of 1,980 buildings in Gangnam-gu, Seongdong-gu, and Yongsan-gu, Seoul, South Korea for 2019. The basic building information comprises 10 categories and is divided into continuous and categorical data types. Table 2 lists the names, data types, and brief descriptions of the basic building information used in this study.

The data-preparation phase is crucial for aligning the data with the analytical objectives before the analysis begins. We performed data preprocessing, cleaning, and exploratory data analysis (EDA) to preserve the original data structure (data-centric check-up data). During preprocessing, categorical variables, such as *building category, representative use, detailed use, land occupancy method, envelope code year, building age classification,* and *total floor area classification*, were converted into

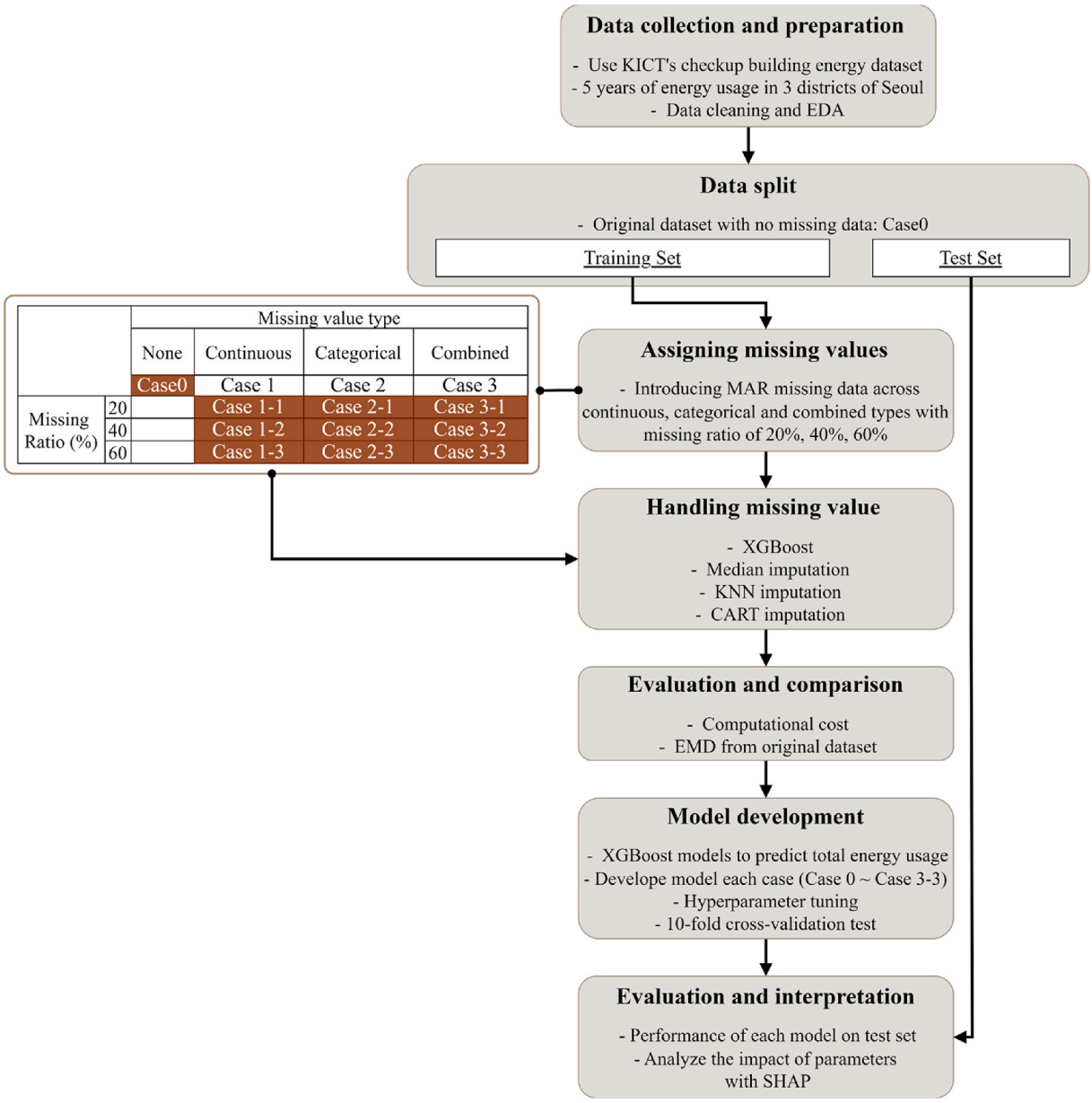


Fig. 1. Research outline.

numerically computable expressions. using label encoding. There are several methods to convert categorical variables into numerical values, including one-hot encoding, label encoding, and target encoding. However, previous studies have shown that the encoding method does not significantly impact the performance of the XGBoost model, which is based on decision trees [33–35]. Therefore, this study employs the commonly used label encoding method.

In label encoding, categorical variables other than *detailed use*, were transformed into numerical data by assigning a specific number to each category. For the 24 types of *detailed use*, a specific number was randomly assigned to each category [36]. This step allowed each category to be converted into independent features, facilitating their inclusion in computational analysis.

In the data cleaning phase, outliers based on total (electric + gas) energy consumption were identified and removed using the interquartile range (IQR) rule, specifically employing the 3IQR method, which eliminated values beyond median ± 1.5IQR. After removing the outliers, 335 buildings from the original 1,980 were excluded based on this criterion; therefore, 1,645 buildings were used to create Case 0. The distribution changes in the total energy consumption before and after outlier removal can be observed in the post-removal distribution (Fig. 2 (b)). This operation was executed to eliminate extreme observations that could potentially distort the analytical outcomes, thereby augmenting the empirical robustness and precision of the subsequent findings.

During the EDA phase, the relationships between the variables were determined using Pearson's correlation analysis [37]. This process is a crucial step that precedes sensitivity and data analyses. Skipping this step and proceeding with the analysis may result in unreliable results [38]. The Pearson correlation coefficient measures the linear relationship between two variables and ranges from −1 to +1, where +1 indicates a perfect positive correlation, 0 indicates no correlation, and −1 indicates a perfect negative correlation. The calculation divided the covariance $\boldsymbol{cov}(\boldsymbol{X}, \boldsymbol{Y})$ between two variables $\boldsymbol{X}, \boldsymbol{Y}$ by the product of their respective standard deviations $(\boldsymbol{s}(\boldsymbol{X}), \boldsymbol{s}(\boldsymbol{Y}))$, as follows:

$$r = \frac{cov(X, Y)}{s(X) \times s(Y)} \tag{1}$$

Table 2
Overview of variables for building energy benchmarking model.

Variable name	Variable type	Example	Description
Building category	Categorical	Single, Complex	Classification system for types of buildings
Building approval year	Continuous	1997 (yr)	Year the building was approved for use
Total floor area	Continuous	1566.65 (m^2)	Total area of the building
Total floor area for FAR calculation	Continuous	1338.02 (m^2)	Floor area to calculate floor area ratio (FAR), which excludes spaces such as basements or parking areas
Representative use	Categorical	Office, Multi-family, Education, Residential neighborhood facilities, Single-family, Accommodation, Retail	Primary use of the building
Detailed use	Categorical	General office, Highschool …	Subcategories of building usage
Land occupancy method	Categorical	Single, Campus	How buildings take up land
Envelope code year	Categorical	~1981 (yr), ~1985 (yr), ~1988 (yr), ~2002 (yr), ~2009 (yr), ~2011 (yr), 2012 (yr) ~	Building's insulation code base year
Building age classification	Categorical	Very old, Old, Moderate, New, Very new	Classification based on building age; as the age of the building increases, the numerical value assigned for its classification also increases
Total floor area classification	Categorical	Very small, Small, Medium, Large, Very large	Classification based on total floor area; as the total floor area of the building increases, the numerical value assigned for its classification also increases

* Year (yr).

2.2. Data split

Before assigning missing values to the Case 0 dataset, which contained no missing data, the data were split into training and test sets in a ratio of 0.7:0.3. The training set was used to generate and replace missing data, create models, and evaluate model stability through 10-fold cross-validation, whereas the test set was utilized to assess the model performance based on the missing data handling method. This separation ensured reliable test data availability, as dividing the data after handling the missing data could alter the characteristics of the test set, potentially impacting the model performance evaluation. This strategy was chosen to secure a trustworthy test set for evaluating the impact of missing data handling methods on model performance before introducing the missing data.

2.3. Assigning missing value

Missing data refers to instances in which values are absent from a dataset, and there are several common reasons for missing data:

1) Historical incompleteness: Older buildings may lack comprehensive records owing to the unavailability of detailed documentation practices in the past. For instance, buildings constructed before the 1980s often have incomplete or missing permit information.
2) Data entry errors: Human errors during data entry can lead to missing or incorrect entries. This is particularly prevalent in large datasets where manual data handling is involved.
3) Changes in data collection standards: Over time, the standards and methods for data collection can evolve. Buildings that were assessed under older standards might have data fields that are not applicable or were not recorded in the past.
4) Privacy and confidentiality concerns: Certain sensitive information might be intentionally omitted to protect privacy. For example, specific details about building ownership might be restricted in public datasets.
5) Physical damage or loss: Physical records can be damaged or lost owing to disasters, poor storage conditions, or other unforeseen circumstances.

In the case of the Korean Building Register, which was primarily used in this study, missing data occurred owing to incorrect entries by building managers, historical recording practices, or the addition of new survey items that were not previously collected.

Handling missing data in an analysis is crucial because it significantly influences the outcomes. Removing all the missing data can

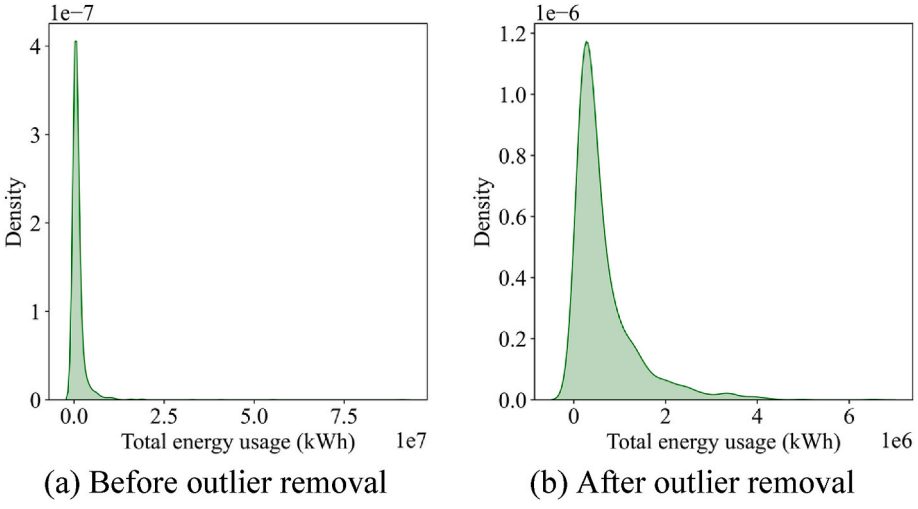


Fig. 2. Distribution of total energy usage after data cleaning.

distort the original data distribution and lead to significant information loss, whereas incorrect imputation can introduce bias. Thus, selecting an appropriate method for handling missing data is crucial for accurate data analysis.

Missing data can be broadly categorized into three types: MCAR, MAR, and missing not at random (MNAR) [15]. Most conventional methods for handling missing data assume that the missing data can be predicted using other variables, which aligns with the MAR principle. However, many studies tend to consider missing data as MCAR without performing an analysis to determine the reasons for missingness, which often does not reflect reality. The MCAR assumption, where missing data are completely independent of other parts of the data, can only realistically occur because of computational errors or omissions in data entry. This assumption is less applicable to building data, where the likelihood of missing MCAR data is particularly low. In building data, there exist close relationships between data points owing to changes over time, regulatory changes, and the consequent close correlations between variables [38]. Therefore, the assumption that the missing data are completely independent of other variables is unrealistic.

This study introduced MAR data, reflecting building data characteristics, to offer a realistic approach for handling missing data in building energy model construction. Fig. 3 shows the permit years of the three districts in Seoul used in the study. The orange distribution represents the *building approval years* of all buildings, while the blue distribution represents the *building approval years* of buildings excluding those with missing values in the dataset. It can be observed that buildings without missing values are concentrated in the more recent permit years. Therefore, this study assumes that older buildings are more likely to have missing data, increasing the probability of missing data linearly with the *building approval year*.

The method for assigning missing data, shown in Fig. 4, introduced missing data into the Case 0 dataset with different missing ratios by variable type to identify the optimal missing data handling method for various missing scenarios. Cases were differentiated by the variable type affected: Case 1 for continuous variables, Case 2 for categorical variables, and Case 3 for all variables, with missing ratios of 20 %, 40 %, and 60 %. The cases increase numerically (e.g., Case 1-1, Case 1–2, Case 1–3) as the missing data ratio escalates, allowing for a structured approach to evaluate missing data handling strategies across variable types.

2.4. Handling missing values

In this study, four imputation methods were compared: XGBoost, Median, KNN, and CART, all of which can be applied to continuous and categorical variables. These four methods were applied to each case (Case 1, Case 1-1, … Case 3-3), and the description of each method is as follows:

XGBoost: Gaining recent attention for its superior performance [25, 39], XGBoost does not replace missing data but utilizes the sparsity-aware split-finding algorithm within the XGBoost framework to manage missing data. The sparsity-aware split-finding algorithm in XGBoost splits data points with missing values and learns to assign the direction (left or right node) of the data with missing values. This method can be applied to both continuous and categorical data without additional steps. The detailed pseudocode for the sparsity-aware split--finding algorithm is provided below [35].

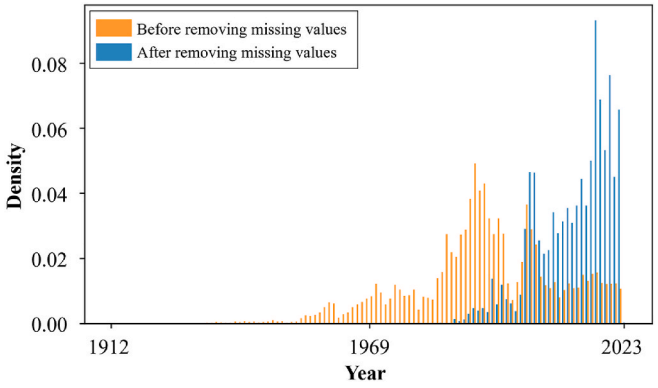


Fig. 3. Distribution of building approval years before and after removing missing values.

Algorithm 1: Sparsity-aware split finding

Input: I, instance set of current node
Input: Ik = { $i \in I | x_{ik} \neq missing$}
Input: d, feature dimension
Applies also to approximate setting; only collects statistics of non-missing entries into buckets
gain ← 0
G ← $\sum_{i\in I} g_i$, H ← $\sum_{i\in I} h_i$
for k = 1 to m do
  // enumerate missing value go to right
  $G_L \leftarrow 0$, $H_L \leftarrow 0$
  for j in sorted (I_k, descent order by x_{jk}) do
    $G_L \leftarrow G_L + g_j$, $H_L \leftarrow H_L + h_j$
    $G_R \leftarrow G - G_L$, $H_R \leftarrow H - H_L$
    $score \leftarrow \max(score, \frac{G_L^2}{H_L+\lambda} + \frac{G_R^2}{H_R+\lambda} - \frac{G^2}{H+\lambda}$
  ends
  // enumerate missing value go to left
  $G_R \leftarrow 0$, $H_R \leftarrow 0$
  for j in sorted (I_k, descent order by x_{jk}) do
    $G_R \leftarrow G_R + g_j$, $H_R \leftarrow H_R + h_j$
    $G_L \leftarrow G - G_R$, $H_L \leftarrow H - H_R$
    $score \leftarrow \max(score, \frac{G_L^2}{H_L+\lambda} + \frac{G_R^2}{H_R+\lambda} - \frac{G^2}{H+\lambda}$)
  end
end
Output: Split and default directions with max gain

The Median method is a single-imputation technique that replaces all missing values with the median, offering the advantage of fast computation time but with the drawback of potentially severely distorting the variable distribution [40]. Similarly, KNN is a single-imputation method and a classic example of the Hot Deck method [41]. The hot deck method involves replacing missing data points with observed values from similar data points. Specifically, the KNN method identifies the nearest neighbors based on the Euclidean distance to the data point with missing values and uses their average (for continuous variables) or mode (for categorical variables) to impute the missing values [39]. Although KNN has shown superior performance in handling missing data for continuous variables, its computational time increases with larger datasets [27].

The CART method, developed by Breiman et al. [42] was identified as the most effective approach for handling missing values in categorical variables, according to previous research [30] that compared performance across methods. Similar to the KNN method, a considerable drawback of the CART method is the increased computational time as the dataset size increases. For classification problems, the CART method handles missing data by iteratively splitting the predictor space. The CART method traverses the missing data points down the tree structure to the most appropriate leaf and extracts a hypothetically appropriate imputation from among the observations belonging to that leaf. For example, in a tree considering variables such as gender, race, and ethnicity (Fig. 5), missing Hispanic males would be imputed using observations from L5 [30]. This study utilized the refined imputation approach developed by Burgette & Reiter [43] by applying Bayesian bootstraps to each leaf to obtain more precise replacement values.

2.5. Evaluation and comparison

This study evaluated missing data handling methods using two criteria. The first criterion is the calculation time required for missing

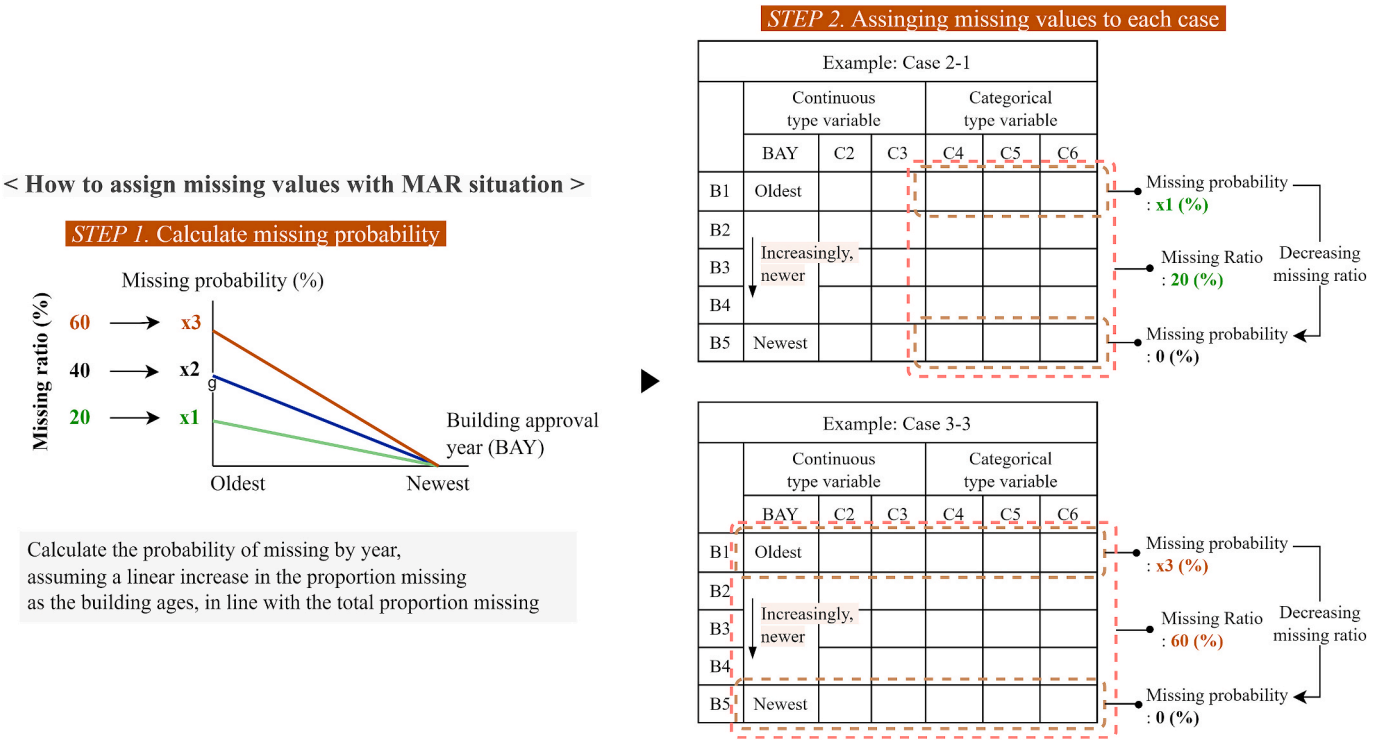


Fig. 4. The concept of assigning missing values with MAR scenarios.

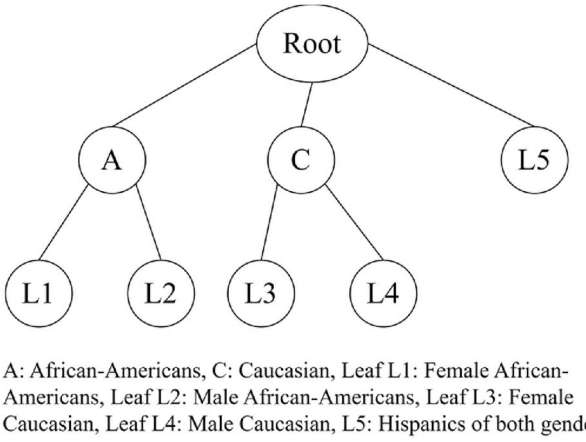


Fig. 5. Example illustration of CART tree structure [43].

data processing, which is a critical factor when dealing with large datasets because excessive processing time can hinder practical analysis or model construction. Therefore, developing an effective and time-efficient method is crucial. The measurements were conducted on a desktop running Windows 11, featuring an AMD Ryzen 9 7950X3D 16-Core Processor, 128 GB RAM, and an NVIDIA GeForce RTX 3060 graphics card. The missing data replacement process utilized software based on Python 3.9.7 with scikit-learn 1.3.0, and the CART method used software based on R 4.2.0 with the MICE 3.16.0 package [44].

The second criterion is the Earth mover's distance (EMD) between the original and imputed data [45]. EMD measures the similarity between two distributions, indicating the minimum cost needed to transform one distribution into another, thus assessing how closely the imputed values match the original data distribution [46]. EMD is calculated using a set of suppliers $\boldsymbol{I}$ and a set of consumers $\boldsymbol{J}$. The EMD of sets $\boldsymbol{I}$ and $\boldsymbol{J}$ is given as follows:

$$EMD = \frac{\sum_{i=1}^{I}\sum_{j=1}^{J} C_{ij} f_{ij}}{\sum_{i=1}^{I}\sum_{j=1}^{J} f_{ij}} \tag{2}$$

where $\boldsymbol{i}$ is a unit in $\boldsymbol{I}$, $\boldsymbol{j}$ is a unit in $\boldsymbol{J}$, $\boldsymbol{C_{ij}}$ is the cost of transporting a unit from $\boldsymbol{I}$ to $\boldsymbol{J}$, and $\boldsymbol{f_{ij}}$ is the flow path.

EMD is a method measuring the distance between two different probability distributions and is useful for evaluating the resemblance of the data distribution after missing data treatment to the original data. A smaller EMD indicates that the post-missing data treatment is closer to the original data. Utilizing the EMD characteristics, this study employed EMD as an indicator to evaluate the ability of the missing data handling methods to preserve the original data characteristics. These evaluations help compare the time efficiency and data characteristic preservation ability of each method, aiding in the selection of the most suitable missing data handling method for building energy model construction.

2.6. Model development

To compare the predictive performance of the model across different missing data scenarios and imputation methods, we constructed models for nine missing data cases, each treated with four different imputation techniques, for a total of 36 scenarios. Models were built using the XGBoost algorithm, with the training set divided into training and test sets in a 0.7:0.3 ratio for hyperparameter tuning.

XGBoost, a machine learning algorithm built on GBDT, was enhanced by Friedman [47] and further improved by Chen and Guestrin [35]. It utilizes sequential decision trees to learn from previous tree errors by summing the scores of each tree for the outcome. Key improvements in XGBoost include more accurate estimates through the second-order Taylor expansion of the loss function and model complexity control with L1 and L2 regularization, which helps prevent overfitting, a common issue in GBDT [2].

During the machine learning model training process, improving and

stabilizing model performance necessitates hyperparameter tuning, in which users adjust the algorithm parameters. Common tuning methods include random search, grid search, and Bayesian optimization. This study employed a grid search to explore all hyperparameter combinations within a predefined range to select the combination that yielded the best performance [48]. To prevent model overfitting during hyperparameter tuning, a 5-fold cross-validation was applied. The descriptions, ranges, and results of the tuned hyperparameters are presented in Tables 3 and 4. Additionally, a 10-fold cross-validation was performed to assess model stability. This step briefly evaluated the consistency of the model predictions across the datasets processed for missing data [49].

2.7. Evaluation and interpretation

To objectively evaluate the final model performance using the test set, the process from data split to model development was repeated 10 times by changing the seed number. The seed number refers to the starting point used in a random number generator. By changing the seed number, the data-splitting process can be varied and repeated to objectively evaluate the model performance. R-squared (R^2), which is commonly employed to evaluate building energy benchmarking model performance [7,50], was used as the performance metric. R^2 is a statistic used in regression analysis that indicates how well the model explains the actual data. R^2 was calculated as follows:

$$R^2 = 1 - \frac{SS_{res}}{SS_{tot}} = 1 - \frac{\sum_{i=1}^{n}\left(y_{predict,i} - y_{data,i}\right)^2}{\sum_{i=1}^{n}\left(y_{data,i} - \bar{y}_{data}\right)^2} \quad (3)$$

where SS_{res} is the sum of the squared residuals, that is, the sum of the squared differences between the actual and predicted values, and SS_{tot} is the difference between the actual and mean values of all data.

For model interpretation, tree-based machine learning algorithms, such as XGBoost, provide feature importance results but lack support for interpreting whether each feature positively or negatively correlates with the target value. Because building energy benchmarking models must be precise and allow for accurate interpretation, this study employed the SHAP method for model interpretation. SHAP is a game-theoretic approach to explain model outcomes, representing an XAI-based analysis method [51]. Unlike conventional global sensitivity analysis, which quantitatively represents the impact of each feature on the target value [52], the SHAP method offers the advantage of intuitively understanding the magnitude and direction of the influence of individual points within a feature on the target value through SHAP values.

Table 3
Hyperparameter tuning range.

Hyperparameter	Description	Range
Learning rate	Rate at which the model learns	0.01–0.1
Estimators	Number of trees in the ensemble	50–1000
Max depth	Maximum depth of a tree. Increasing this value makes the model more complex and likely to overfit	1–10
Min child weight	Minimum sum of instance weight (hessian) needed in a child	2–10
Subsample	Subsample ratio of the training instance	0.1–1
Colsample bytree	Subsample ratio of columns when constructing each tree	0.1–1
Colsample bynode	Subsample ratio of columns from each node	0.1–1
Colsample bylevel	Subsample ratio of columns for each split, in each level	0.1–1

Table 4
Hyperparameter tuning results.

Hyperparameter	Value
Learning rate	0.1
Estimators	60
Max depth	3
Min child weight	10
Subsample	1
Colsample bytree	0.7
Colsample bynode	1
Colsample bylevel	0.6

3. Results and discussion

3.1. Correlation analysis results

The correlation analysis between variables needed for building an energy benchmarking model, as shown in Fig. 6, indicates a very high positive correlation ($\gamma = 0.96$) between *total floor area* and *total floor area for FAR calculation*. This correlation exists because the *total floor area for FAR calculation* excludes areas not counted in the floor area ratio (FAR), such as basements and parking lots, meaning that larger total floor areas naturally lead to higher values for this variable. The high positive correlation between the *total floor area* ($\gamma = 0.87$) and *the total floor area for FAR calculation* ($\gamma = 0.84$) reflects a higher *total floor area classification*.

A very high positive correlation ($\gamma = 0.92$) was found between *building approval year* and *envelope code year*, reflecting stricter insulation standards in more recently constructed buildings. Conversely, as the *building approval year* approaches the present, the *building age classification* decreases, leading to a very high negative correlation ($\gamma = -0.96$) between these two variables, indicating that newer buildings are classified as having a lower age.

The observed strong negative correlation ($\gamma = -0.89$) between the *envelope code year* and *building age classification* underscores a distinct trend. This reflects an inverse relationship, in which advancements in envelope codes for newer constructions lead to lower age classifications. This analysis of the correlations among the various features used in constructing building energy benchmarking models aids in understanding how these characteristics interrelate and their impact on the energy benchmarking model.

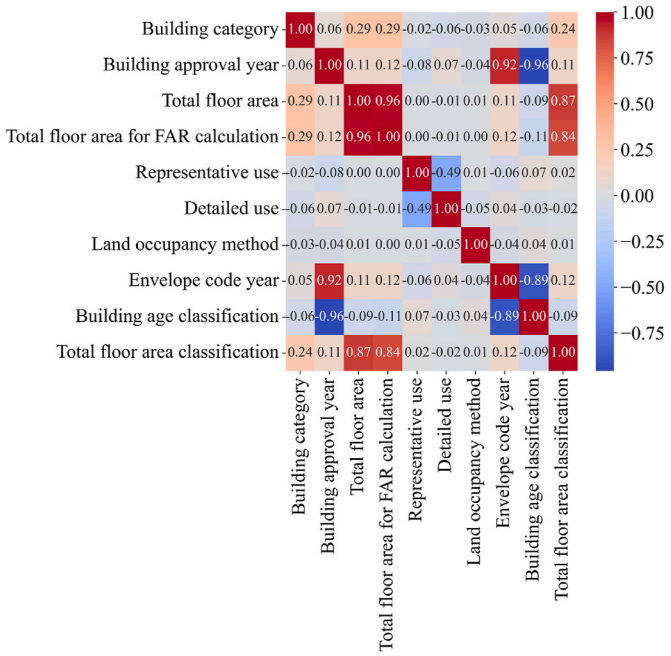


Fig. 6. Case 0 correlations analysis results.

3.2. Model interpretation: SHAP method

Fig. 7 shows the sensitivity analysis of each variable in the model created using Case 0 (reference data without missing values) and the SHAP summary plot obtained using the SHAP method. The X-axis represents the SHAP values, where positive and negative values indicate positive and negative correlations between the input variables and total energy consumption (kWh), respectively. The Y-axis shows the names of the variables sorted by importance, with blue and red dots indicating smaller and larger values within each variable, respectively.

The analysis reveals that the three area-related variables (*total floor area, total floor area for FAR calculation,* and *total floor area classification*) were the most crucial in the model, showing a positive correlation with total energy consumption. This indicates that larger areas lead to higher energy use. This result may have occurred because the *total floor area* of the building is pivotal in energy consumption or from interaction effects from the highly variable correlations observed in Section 3.1. The next most important variable is *detailed use*. Fig. 7 shows that total energy use increases as the value of *detailed use* increases. However, this cannot be interpreted as a positive correlation given the variable nature. This pattern appears to be an incidental consequence of the label encoding technique, which assigns sequential numbers to the categorical variable *detailed use*, and is not an indicator of energy consumption trends. For the *building approval year*, a negative correlation was observed, indicating that newer buildings tend to have lower energy consumption, which aligns with the understanding that recent constructions are more energy efficient. Other variables showed no distinct correlation with total energy usage.

The results show that the SHAP method, an XAI method, enables interpretable building energy benchmarking models, assisting stakeholders in comprehending the decision-making process and effectively applying outcomes.

3.3. Missing data handling method evaluation: calculation costs and EMD

Fig. 8 compares the time required for different missing data handling methods. The imputation time refers to the time spent on missing data treatment alone, while the total time includes both the imputation and subsequent model training time, where the model training time was consistently 0.75 s. The model training time of the XGBoost method was 0 s, as it skips the imputation process. The Median and KNN methods had relatively lower times of 0.01 and 0.37 s, respectively, indicating their efficiency. The CART method took 578.83 s, which is almost equal to the total time of 579.59 s, highlighting that most of the time was spent on handling missing data.

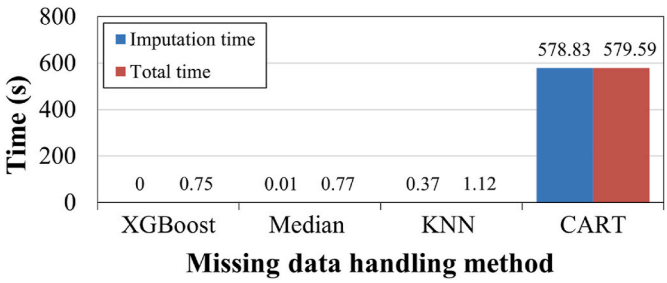


Fig. 8. Calculation time for different missing data handling methods , and show the changes in the EMD between the original data and the data treated with various missing data handling methods across different missing scenarios. Notably, for the XGBoost method, the EMD was calculated solely on the existing data to measure the distribution difference from the original data. This is because the XGBoost method does not fill in missing values with any specific value, making it impossible to measure the distribution in the presence of missing data. Therefore, based on the EMD calculations, the XGBoost method can be considered similar to the removal of missing values.

This indicates that the missing data handling method used during the data preprocessing phase can significantly alter the total construction time of the energy benchmarking model, and that model construction time is particularly affected by large datasets and limited computational resources. Thus, it is crucial for practitioners to select the optimal method for handling missing data.

Table 5 shows the EMD results for the missing values for Cases 1-1, 1–2, and 1–3, where only continuous variables have missing values. Table 5 compares the EMD values for *building approval year*, *total floor area*, and *total floor area for FAR calculation* according to the missing data handling methods. The CART method had lower EMD values than the KNN and Median methods in all cases. This indicates that the CART method outperforms the other methods in accurately replicating the original data distribution. The KNN method maintained a distribution similar to that of the original data. For the Median method, the EMD was

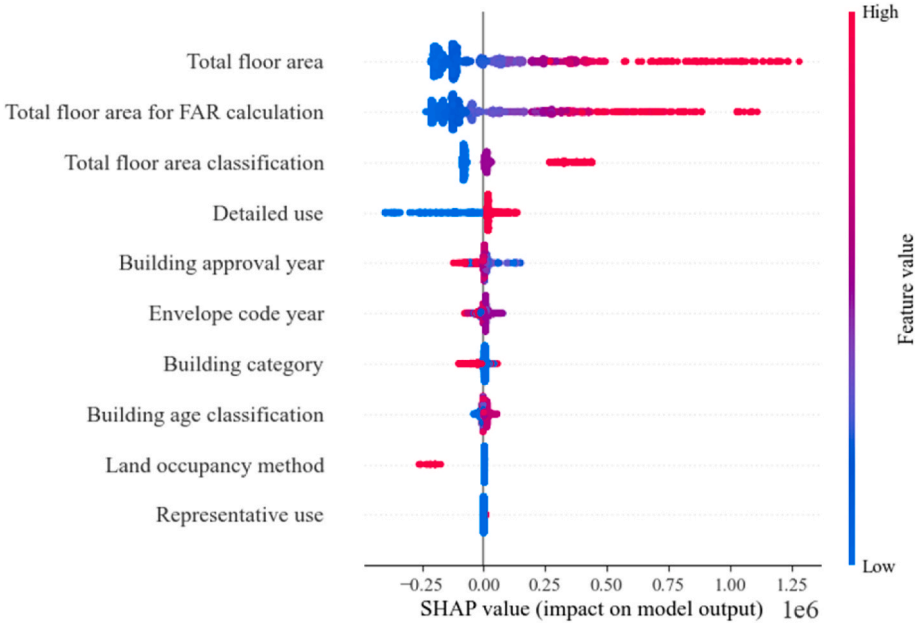


Fig. 7. SHAP summary plot of feature attributions generated by Case 0 model.

Table 5
EMD results for Case 1 (continuous variable type missing cases).

	Missing ratio (%)	Variable	XGBoost	Median	KNN	CART
Case 1-1	20 %	Building approval year	1.30	1.79	0.87	0.11
		Total floor area	69.85	539.17	67.94	37.68
		Total floor area for FAR calculation	59.25	347.15	51.65	33.70
Case 1–2	40 %	Building approval year	3.46	4.89	2.25	0.32
		Total floor area	147.36	1045.75	138.06	80.09
		Total floor area for FAR calculation	127.91	700.10	94.78	51.93
Case 1–3	60 %	Building approval year	6.95	7.95	4.69	1.38
		Total floor area	255.52	1546.52	181.80	106.91
		Total floor area for FAR calculation	242.33	1038.35	174.10	99.93

higher than that of the XGBoost method, indicating that the original data distribution was severely distorted. The EMD distance also increased as the percentage of missing data increased. For example, for the *building approval year* in the 20 % missing scenario, the EMD for the Median, KNN, and CART methods was 1.79, 0.87, and 0.11, respectively, but as the percentage of missing data increased to 40 %, these values increased to 4.89, 2.25, and 0.32, respectively. This suggests that as the proportion of missing data in the original data increases, the ability of an imputation method to reproduce the original data distribution decreases.

The EMD results for Case 2, in which only the categorical variables are missing, are presented in Table 6. Similar to the results in Table 5, where only continuous variables are missing, the CART method generally reproduces the original data distribution better than the Median and KNN methods. Both the KNN and CART methods have lower EMD values overall. In addition, the Median method has a higher EMD than the XGBoost method, making it the least effective among the imputation methods. Furthermore, the EMD tended to increase as the proportion of missing values increased. Similar to the EMD results for continuous variables, this indicates that as the proportion of missing values in the original data increased, the ability of the methods to reproduce the original data distribution decreased.

Table 6
EMD results for Case 2 (categorical variable type missing cases).

	Missing ratio (%)	Variable	XGBoost	Median	KNN	CART
Case 2-1	20 %	Building category	0.01	0.03	0.01	0.01
		Representative use	0.01	0.02	0.01	0.01
		Detailed use	0.06	0.20	0.09	0.05
		Land occupancy method	0.00	0.00	0.00	0.00
		Envelope code year	0.17	0.17	0.09	0.00
		Building age classification	0.24	0.33	0.14	0.00
		Total floor area classification	0.02	0.12	0.01	0.00
Case 2-2	40 %	Building category	0.01	0.06	0.02	0.01
		Representative use	0.02	0.04	0.02	0.01
		Detailed use	0.13	0.37	0.17	0.12
		Land occupancy method	0.00	0.00	0.00	0.00
		Envelope code year	0.46	0.45	0.27	0.00
		Building age classification	0.62	0.90	0.37	0.01
		Total floor area classification	0.03	0.24	0.02	0.00
Case 2–3	60 %	Building category	0.02	0.09	0.03	0.01
		Representative use	0.02	0.06	0.03	0.04
		Detailed use	0.23	0.53	0.30	0.25
		Land occupancy method	0.00	0.01	0.00	0.01
		Envelope code year	0.88	0.89	0.54	0.17
		Building age classification	1.24	1.29	0.78	0.06
		Total floor area classification	0.05	0.37	0.03	0.00

The results for Case 3, involving both continuous and categorical variables with missing data, are presented in Table 7. In most instances, the CART method replicated the original data distribution closely after imputation, followed by the KNN method. Similar to previous findings, the Median method showed the greatest deviation from the original data distribution. An increase in the proportion of missing data led to an increase in the EMD, indicating the challenges of accurately replicating the original data distribution with higher rates of missing data.

Considering the computation time and EMD results, the CART method consistently yielded datasets that were closest to the original data distribution across all missing data scenarios. However, because the CART method requires the longest time to impute missing data, the KNN method can be used for imputing missing data when there are computation time constraints. For cases with missing data only for categorical variables, both the KNN and CART methods achieved distributions similar to those of the original dataset. The XGBoost method generally falls between the KNN and Median methods in terms of producing datasets with distributions similar to the original data. The Median method, which significantly distorted the original data distribution, is not recommended for constructing building energy models.

In addition, removing missing data can lead to unintended distortions in the dataset, which is not advisable, particularly for building data with older buildings, which are more likely to have missing data. This tendency could bias the dataset toward newer buildings, potentially underrepresenting older ones, which may affect the accuracy of building energy benchmarking models for older structures. Fig. 9 shows the density distribution of approved years as a function of the percentage of missing data. Case 0 represents the original dataset with no missing data, while Case 3-3 represents cases in which 60 % of the data are missing. From this plot, we can see that the overall distribution of Case 3-3 is different from that of Case 0. In particular, Case 0 had a peak in the *building approval year* between 1980 and 2000, whereas Case 3-3 had a peak between 2000 and 2020. This indicates that removing missing values to create a dataset may skew the dataset toward newer buildings and may not accurately reflect the actual age distribution of the buildings.

3.4. Model performance based on missing data handling method

The predictive performance of the models under various missing data scenarios based on the missing data handling method used was evaluated. Figs. 10–12 how the performance changes for models with missing data for continuous, categorical, and all variables, respectively. Fig. 13 presents a comprehensive comparison of model performance across all missing data scenarios. The performance changes were analyzed using: (a) the results of 10-fold cross-validation on the training set with missing data, and (b) the model performance results using the test set.

The plots in Figs. 10–12 can be read as follows. The X-axis represents the percentage of missing data (20 %, 40 %, and 60 %), with each data

Table 7
EMD results for Case 3 (continuous + categorical variable-type cases).

	Missing ratio (%)	Variable	XGBoost	Median	KNN	CART
Case 3-1	20 %	Building category	0.00	0.03	0.02	0.00
		Building approval year	1.30	1.60	0.89	0.16
		Total floor area	69.85	486.68	69.85	33.58
		Total floor area for FAR calculation	59.25	308.72	52.06	34.34
		Representative use	0.01	0.02	0.02	0.01
		Detailed use	0.06	0.19	0.16	0.06
		Land occupancy method	0.00	0.00	0.00	0.00
		Envelope code year	0.17	0.18	0.11	0.01
		Building age classification	2.22	0.32	0.16	0.02
		Total floor area classification	0.01	0.12	0.01	0.00
Case 3-2	40 %	Building category	0.01	0.07	0.03	0.01
		Building approval year	3.46	4.37	2.26	1.07
		Total floor area	147.36	943.75	129.10	72.23
		Total floor area for FAR calculation	127.91	622.98	103.30	51.34
		Representative use	0.02	0.04	0.04	0.02
		Detailed use	0.12	0.37	0.32	0.09
		Land occupancy method	0.00	0.00	0.00	0.00
		Envelope code year	0.45	0.50	0.27	0.14
		Building age classification	0.59	0.89	0.38	0.14
		Total floor area classification	0.03	0.25	0.02	0.01
Case 3-3	60 %	Building category	0.02	0.09	0.04	0.01
		Building approval year	6.95	7.13	4.38	3.55
		Total floor area	255.52	1392.85	209.54	110.27
		Total floor area for FAR calculation	242.33	931.55	188.95	101.08
		Representative use	0.02	0.06	0.06	0.02
		Detailed use	0.20	0.52	0.44	0.17
		Land occupancy method	0.00	0.01	0.01	0.01
		Envelope code year	0.87	0.89	0.51	0.43
		Building age classification	1.21	1.28	0.82	0.58
		Total floor area classification	0.01	0.36	0.04	0.01

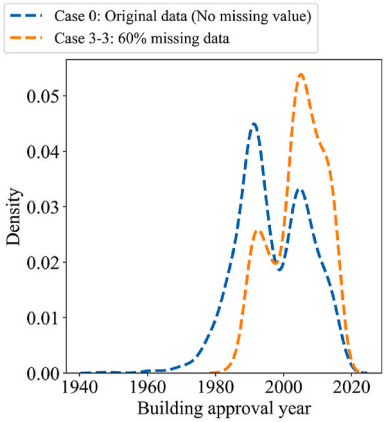
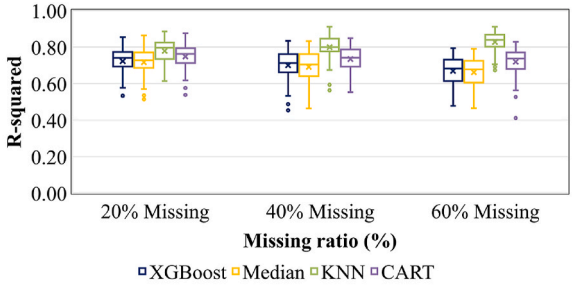
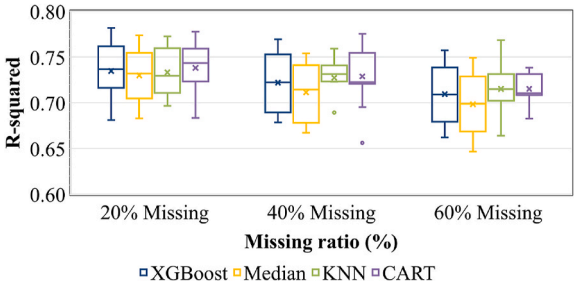


Fig. 9. Building approval year density comparison before and after assigning missing values.



(a) 10-fold cross-validation results for training set



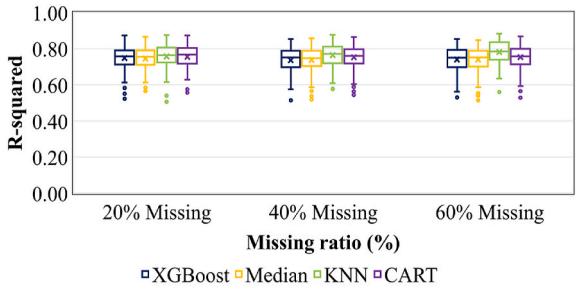
(b) Model performance on test set

Fig. 10. Model performance comparison results of handling missing data in continuous variable types.

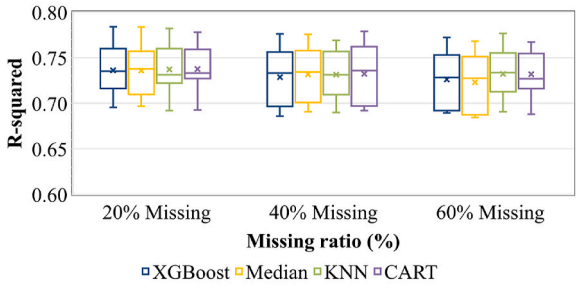
point corresponding to the performance of the models at these missing data levels. The Y-axis represents the R^2 value, indicating the explanatory power of the models, with higher values suggesting better performance. The plots use different colors and symbols to represent the four missing data handling methods: XGBoost (Blue), Median (Yellow), KNN (Purple), and CART (Green). The performance of each method is displayed using box plots, which visualize the distribution of the R^2 values over 10 iterations.

Only continuous variables missing: Fig. 10(a) shows that the KNN and CART methods outperform the XGBoost and Median methods in terms of model performance. Notably, the KNN method shows an increase in model predictive performance as the amount of missing data increases, with further interpretation provided in Fig. 13.

Fig. 10(b) shows that, similar to the results in Fig. 10(a), the CART and KNN methods generally have higher accuracy; in particular, the CART method has a smaller model performance deviation than the KNN method. For the XGBoost method, the model performance deviation was larger than those of the KNN and CART methods; however, the overall performance was not significantly lower. These results suggest that

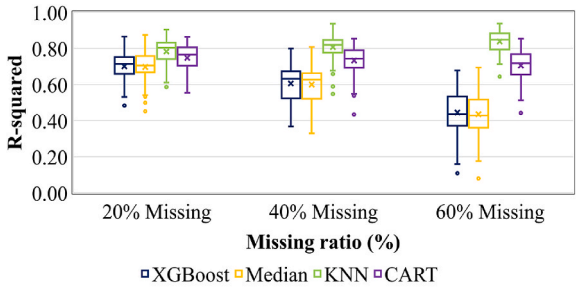


(a) 10-fold cross-validation results for training set

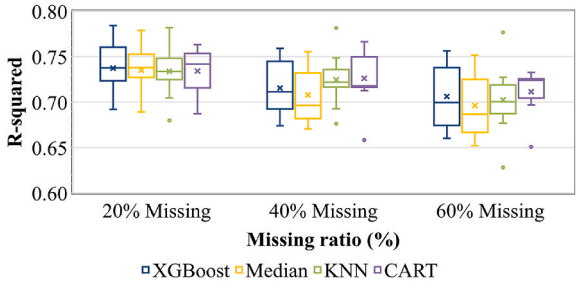


(b) Model performance on test set

Fig. 11. Model performance comparison results of handling missing data in categorical variable types.



(a) 10-fold cross-validation results for training set



(b) Model performance on test set

Fig. 12. Model performance comparison results based on handling missing data in continuous + categorical variable types.

omitting missing values may not significantly reduce the model performance in certain scenarios.

In conclusion, for missing continuous variables, imputation using the CART method is the most stable and performs well in terms of model prediction performance; however, considering the computational time, the XGBoost and KNN methods are also good choices.

Only categorical variables missing: Fig. 11(a) shows that, similar to Fig. 10(a), missing value imputation using the KNN method results in the highest model prediction performance and model performance increases as the percentage of missing values increases. In Fig. 11(b), the CART and KNN methods show a more stable model explanatory power than the XGBoost and Median methods, but there was no significant difference in the model performance. This may be because the difference in model prediction performance between the methods was not significant, as the EMD of the missing categorical variable values was not large compared with that of the original data. These results suggest that missing categorical variable values may have a relatively small impact on model performance.

All variables missing: Fig. 12 shows that the variability in model explainability is greater than in cases where missing data are present in only one variable type. Similar to the other results, as shown in Fig. 12 (a), the KNN method exhibits the best prediction performance, and the model performance improves as the percentage of missing values increases. For the XGBoost and Median methods, the model performance decreased significantly as the percentage of missing values increased, and the model performance deviation tended to increase. In contrast, the CART method showed stable performance, although the explanatory power of the model decreased slightly as the amount of missing data increased.

In Fig. 12(b), the CART method shows the best and least deviating model performance, with the KNN method also performing well. Although the XGBoost method had a larger deviation in model performance than the CART and KNN methods, it still maintained sufficient model prediction performance when it was the only method capable of imputation. The Median method had the lowest performance and the largest model variance. These results indicate that the CART method is the most stable and useful method for building a high-performing model, even when all variables are missing. However, considering the computational time, the XGBoost and KNN methods can also be considered effective.

Comprehensive comparison across all scenarios: Fig. 13 shows a comprehensive comparison of model performance across all missingness scenarios. Comparing Fig. 13(a) and (b), we observe that model performance evaluated with the training set containing missing data exhibits larger deviation than that evaluated with the complete test set. From Fig. 13(a), it is evident that the KNN method outperforms the CART method in all scenarios. Interestingly, the KNN method shows an increase in model performance as the amount of missing data increases, which contrasts with the EMD results in Section 3.3.

These results can be interpreted in two ways. First, similarity to the original data distribution after imputation does not necessarily guarantee improved model prediction performance. Despite the CART method having the smallest EMD distance, theoretically suggesting superior performance, the KNN method demonstrates better overall prediction performance when evaluated with the training set. This improvement with the KNN method may stem from its ability to enhance existing data patterns through similarity-based imputation. By replacing missing values with those from similar data points, the KNN method reinforces existing data patterns, with this effect becoming more pronounced as the proportion of missing values increases. Consequently, model performance improves as missing data increase, as shown in and. Therefore, caution is advised when using the KNN method for imputation to avoid overestimating model performance.

Second, the importance of test data in evaluating building energy model performance is highlighted. If the test data contain missing values, their handling can significantly affect the reliability assessment of the model. The results in Fig. 13(b), which evaluate model performance on the test set without missing data, show relatively small deviation in model performance, with a maximum R^2 of 0.2. In Fig. 13(a),

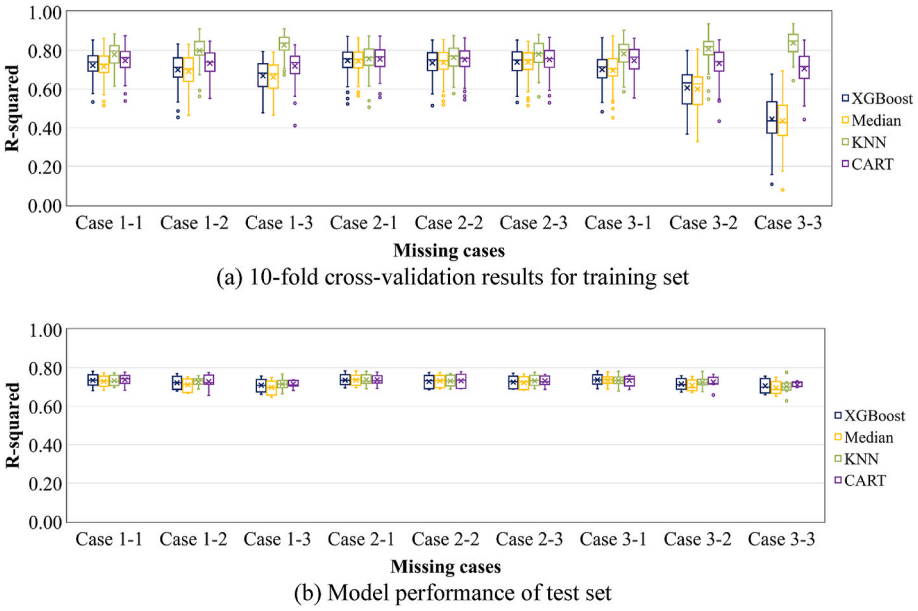


Fig. 13. Model performance comparison for all missing cases.

presenting a 10-fold cross-validation on the training set with missing data, we observe considerable deviation in model performance, with a maximum R^2 of 0.7 depending on how missing data are handled. This disparity indicates that evaluating model performance without appropriate test data can yield unreliable results in assessing predictive power. Therefore, when evaluating the performance of building energy benchmark models, obtaining high-quality test data beforehand is crucial to ensure accurate performance assessment. For example, if the CART method is used for preprocessing missing values in building energy benchmark model construction, and the KNN method is subsequently applied to replace missing values in basic building energy information, the resulting model performance may differ from initial expectations.

4. Conclusions

This study investigated the total computation time, EMD, and model performance of different missing data handling methods for building energy benchmarking models. Unlike previous studies that focused on the missing target values of the model, this study emphasized the input data quality and applied the MAR assumption to the missing data scenario, which realistically reflects the characteristics of building data and missing data scenarios compared to the MCAR assumption used previously.

The results indicated that the CART method required significantly more computing time but produced a dataset with a distribution most similar to the original data. The KNN method had the second-lowest EMD, followed by XGBoost and Median, which distorted the original data distribution the most. Therefore, the CART method is recommended when producing datasets where maintaining a distribution similar to the original data is crucial, while the KNN method provides an efficient alternative when computational resources are limited.

Model performance comparisons indicated that the CART method had the best performance in all missingness scenarios in the test set with no missing data, showing the smallest performance deviation. The KNN method also performed well, while the XGBoost method, despite its larger bias, remained consistent and served as a viable alternative when computational time was limited. For missing categorical data, the CART method is not recommended owing to its long computational time and performance similar to other methods. By utilizing a training set with missing values to conduct 10-fold cross-validation, the analysis confirmed that the KNN method exhibited the best model performance. Moreover, the model performance improved as the number of missing values increased, likely because the KNN method reinforces existing data patterns, which is amplified with higher percentages of missing values. These results highlight important methodological implications for research on missing data, underscoring the necessity of reliable test data for accurately assessing model performance.

The results of applying the provided methods to other datasets with different input data may vary depending on several factors, including the nature of the data, specific characteristics of the datasets, and context of the application. Our method demonstrated robust performance on building energy-related datasets in Korea, suggesting its potential utility in similar contexts. However, it is important to recognize that each dataset may present unique challenges and variations that could affect the outcomes. To address this, we plan to conduct additional studies on diverse datasets to evaluate the generalizability and adaptability of our method.

Despite these limitations, the study provides practical guidelines for selecting imputation methods in the data preprocessing stage of building energy benchmarking model construction and highlights the importance of reliable test data for accurate performance measurement. Future research will aim to consider a wider variety of basic building information and develop more accurate and efficient benchmarking models by securing better datasets, ultimately contributing to increased building energy efficiency and reduced carbon emissions.

Funding

This work was supported by the Korean Ministry of Trade, Industry, and Energy, Republic of Korea [grant number RS-2023-00237035], and the Korean Ministry of Land, Infrastructure, and Transport [grant number RS-2023-00244769].

CRediT authorship contribution statement

Kyungjae Lee: Writing – original draft, Software, Methodology. **Hyunwoo Lim:** Writing – review & editing, Supervision, Conceptualization. **Jeongyun Hwang:** Validation, Formal analysis. **Doyeon Lee:** Visualization, Software.

Declaration of competing interest

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

Data availability

The authors do not have permission to share data.

References

[1] World energy outlook 2022. 2022.
[2] Liu X, Tang H, Ding Y, Yan D. Investigating the performance of machine learning models combined with different feature selection methods to estimate the energy consumption of buildings. Energy Build 2022;273. https://doi.org/10.1016/j.enbuild.2022.112408.
[3] O'Brien W, Tahmasebi F, Andersen RK, Azar E, Barthelmes V, Belafi ZD, et al. An international review of occupant-related aspects of building energy codes and standards. Build Environ 2020;179. https://doi.org/10.1016/j.buildenv.2020.106906.
[4] Huovila P, Ala-Juusela M, Melchert L, Pouffary S, Cheng C-C, Ürge-Vorsatz D, et al. Buildings and climate change: summary for decision-makers. 2009.
[5] Energy Star E. DataTrends energy use benchmarking. 2022.
[6] Nguyen A-T, Reiter S, Rigo P. A review on simulation-based optimization methods applied to building performance analysis. Appl Energy 2014;113:1043–58. https://doi.org/10.1016/j.apenergy.2013.08.061.
[7] Amasyali K, El-Gohary NM. A review of data-driven building energy consumption prediction studies. Renew Sustain Energy Rev 2018;81:1192–205. https://doi.org/10.1016/j.rser.2017.04.095.
[8] Star E. Benchmark your building using ENERGY STAR® portfolio manager®, vol. 4; April 2022.
[9] Papadopoulos S, Kontokosta CE. Grading buildings on energy performance using city benchmarking data. Appl Energy 2019;233–234:244–53. https://doi.org/10.1016/j.apenergy.2018.10.053.
[10] Arjunan P, Poolla K, Miller C. EnergyStar++: towards more accurate and explanatory building energy benchmarking. Appl Energy 2020;276:115413. https://doi.org/10.1016/j.apenergy.2020.115413.
[11] Olu-Ajayi R, Alaka H, Sulaimon I, Sunmola F, Ajayi S. Building energy consumption prediction for residential buildings using deep learning and other machine learning techniques. J Build Eng 2022;45:103406. https://doi.org/10.1016/j.jobe.2021.103406.
[12] Pino-Mejías R, Pérez-Fargallo A, Rubio-Bellido C, Pulido-Arcas JA. Comparison of linear regression and artificial neural networks models to predict heating and cooling energy demand, energy consumption and CO 2 emissions. Energy 2017;118:24–36. https://doi.org/10.1016/j.energy.2016.12.022.
[13] Fan C, Yan D, Xiao F, Li A, An J, Kang X. Advanced data analytics for enhancing building performances: from data-driven to big data-driven approaches. Build Simul 2021;14:3–24. https://doi.org/10.1007/s12273-020-0723-1.
[14] Robinson C, Dilkina B, Hubbs J, Zhang W, Guhathakurta S, Brown MA, et al. Machine learning approaches for estimating commercial building energy consumption. Appl Energy 2017;208:889–904. https://doi.org/10.1016/j.apenergy.2017.09.060.
[15] Khademi A. Flexible imputation of missing data 2nd edition. J Stat Software 2020;93:1–4. https://doi.org/10.18637/jss.v093.b01.
[16] Ma J, Cheng JCP, Jiang F, Chen W, Wang M, Zhai C. A bi-directional missing data imputation scheme based on LSTM and transfer learning for building energy data. Energy Build 2020;216:109941. https://doi.org/10.1016/j.enbuild.2020.109941.
[17] Li H, Chen X, Shan M, Duan P. Missing data filling methods of air-conditioning power consumption for public buildings. In: 2020 39th Chinese control conference (CCC). IEEE; 2020. p. 3183–7. https://doi.org/10.23919/CCC50068.2020.9188857.
[18] Liu X, Zhang Z. A two-stage deep autoencoder-based missing data imputation method for wind farm SCADA data. IEEE Sensor J 2021;21:10933–45. https://doi.org/10.1109/JSEN.2021.3061109.
[19] Hussain SN, Aziz AA, Hossen MJ, Aziz NAA, Murthy GR, Mustakim FB. A novel framework based on cnn-lstm neural network for prediction of missing values in electricity consumption time-series datasets. Journal of Information Processing Systems 2022;18:115–29.
[20] Jung S, Moon J, Park S, Rho S, Baik SW, Hwang E. Bagging ensemble of multilayer perceptrons for missing electricity consumption data imputation. Sensors 2020;20:1772. https://doi.org/10.3390/s20061772.
[21] Wijesekara WMLKN, Liyanage L. Comparison of imputation methods for missing values in air pollution data: case study on sydney air. Quality Index 2020:257–69. https://doi.org/10.1007/978-3-030-39442-4_20.
[22] Kim T, Ko W, Kim J. Analysis and impact evaluation of missing data imputation in day-ahead PV generation forecasting. Appl Sci 2019;9:204. https://doi.org/10.3390/app9010204.
[23] Kim J, Kwak Y, Mun S-H, Huh J-H. Imputation of missing values in residential building monitored data: energy consumption, behavior, and environment information. Build Environ 2023;245:110919. https://doi.org/10.1016/j.buildenv.2023.110919.
[24] Fauzan MA, Murfi H. The accuracy of XGBoost for insurance claim prediction. Int J Adv Soft Comput Appl 2018;10:159–71.
[25] Rusdah DA, Murfi H. XGBoost in handling missing values for life insurance risk prediction. SN Appl Sci 2020;2:1336. https://doi.org/10.1007/s42452-020-3128-y.
[26] Bertsimas D, Pawlowski C, Zhuo YD. From predictive methods to missing data imputation: an optimization approach. J Mach Learn Res 2018;18:1–39.
[27] Jadhav A, Pramod D, Ramanathan K. Comparison of performance of data imputation methods for numeric dataset. Appl Artif Intell 2019;33:913–33. https://doi.org/10.1080/08839514.2019.1637138.
[28] Emmanuel T, Maupong T, Mpoeleng D, Semong T, Mphago B, Tabona O. A survey on missing data in machine learning. J Big Data 2021;8:140. https://doi.org/10.1186/s40537-021-00516-9.
[29] Sanjar K, Bekhzod O, Kim J, Paul A, Kim J. Missing data imputation for geolocation-based price prediction using KNN–MCF method. ISPRS Int J Geo-Inf 2020;9:227. https://doi.org/10.3390/ijgi9040227.
[30] Akande O, Li F, Reiter J. An empirical comparison of multiple imputation methods for categorical data. Am Statistician 2017;71:162–70. https://doi.org/10.1080/00031305.2016.1277158.
[31] Poulos J, Valle R. Missing data imputation for supervised learning. Appl Artif Intell 2018;32:186–96. https://doi.org/10.1080/08839514.2018.1448143.
[32] Lee S-E. Data-Centric Checkup Technique of Building Energy Performance. 2023. https://scienceon.kisti.re.kr/srch/selectPORSrchReport.do?cn=TRKO202400000427. [Accessed 2 September 2024].
[33] Bentéjac C, Csörgő A, Martínez-Muñoz G. A comparative analysis of XGBoost 2019. https://doi.org/10.1007/s10462-020-09896-5; 2007.
[34] Poslavskaya E, Korolev A. Encoding categorical data: is there yet anything "hotter" than one-hot encoding?. 2023.
[35] Chen T, Guestrin C. XGBoost: a scalable tree boosting system. In: Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining. New York, NY, USA: ACM; 2016. p. 785–94. https://doi.org/10.1145/2939672.2939785.
[36] Hastie T, Friedman J, Tibshirani R. The elements of statistical learning. New York, NY: Springer New York; 2001. https://doi.org/10.1007/978-0-387-21606-5.
[37] Benesty J, Chen J, Huang Y, Cohen I. Pearson correlation coefficient. In: Springer topics in signal processing. Springer Science and Business Media B.V; 2009. p. 1–4. https://doi.org/10.1007/978-3-642-00296-0_5.
[38] Lee K, Lim H. Correlation analysis of building parameters according to ASHRAE Standard 90.1. J Build Eng 2024;82:108130. https://doi.org/10.1016/j.jobe.2023.108130.
[39] Mustika WF, Murfi H, Widyaningsih Y. Analysis accuracy of XGBoost model for multiclass classification - a case study of applicant level risk prediction for life insurance. In: 2019 5th international conference on science in information Technology (ICSITech). IEEE; 2019. p. 71–7. https://doi.org/10.1109/ICSITech46713.2019.8987474.
[40] Ramli MNN, Yahaya AS, Ramli NA, Yusof NFFM, Abdullah MMA. Roles of imputation methods for filling the missing values: a review. Adv Environ Biol 2013;7:3861+.
[41] Zhang S. Nearest neighbor selection for iteratively kNN imputation. J Syst Software 2012;85:2541–52. https://doi.org/10.1016/j.jss.2012.05.073.
[42] Breiman L, Friedman JH, Olshen RA, Stone CJ. Classification and regression trees. Routledge; 2017. https://doi.org/10.1201/9781315139470.
[43] Burgette LF, Reiter JP. Multiple imputation for missing data via sequential regression trees. Am J Epidemiol 2010;172:1070–6. https://doi.org/10.1093/aje/kwq260.
[44] Nadarajah S, Kotz S. R Programs for computing truncated distributions, vol. 16; 2006.
[45] Rubner Y, Tomasi C, Guibas LJ. A metric for distributions with applications to image databases. In: Sixth international conference on computer vision (IEEE Cat. No.98CH36271). Narosa Publishing House; 1998. p. 59–66. https://doi.org/10.1109/ICCV.1998.710701.
[46] Wang Y, Shangguan Y, Wang Z, Xue Y. The influence and adjust method of hyperparameters' prior distributions in Bayesian calibration for building stock energy prediction. Energy Build 2022;273:112413. https://doi.org/10.1016/j.enbuild.2022.112413.
[47] Friedman JH. Greedy function approximation: a gradient boosting machine. Ann Stat 2001;29. https://doi.org/10.1214/aos/1013203451.
[48] Bergstra J, Bardenet R, Bengio Y, Kégl B. Algorithms for hyper-parameter optimization. Adv Neural Inf Process Syst 2011;24.
[49] Browne MW. Cross-validation methods. J Math Psychol 2000;44:108–32.
[50] Chicco D, Warrens MJ, Jurman G. The coefficient of determination R-squared is more informative than SMAPE, MAE, MAPE, MSE and RMSE in regression analysis evaluation. PeerJ Comput Sci 2021;7:e623. https://doi.org/10.7717/peerj-cs.623.
[51] Lundberg SM, Lee S-I. A unified approach to interpreting model predictions. Adv Neural Inf Process Syst 2017;30.
[52] Wei T. A review of sensitivity analysis methods in building energy analysis. Renew Sustain Energy Rev 2013;20:411–9. https://doi.org/10.1016/j.rser.2012.12.014.