The Impact of Data Preprocessing on Federated Learning for Air Quality Prediction: A Comparative Study of Imputation Methods

Xiao Liu

College of Computer Science and Technology, Taiyuan University of Science and Technology, Taiyuan, China

202320060410@stu.tyust.edu.cn

Abstract. Data preprocessing plays a foundational role in machine learning, but it receives limited systematic attention in federated learning (FL) environments. This study empirically compares four missing value strategies using the UCI dataset: direct deletion, mean imputation, K-Nearest Neighbors (KNN) imputation, and Random Forest (RF) imputation. The setup uses a federated framework with five clients, each running a multi layer perceptron model. The findings show direct deletion achieves the strongest performance, with a mean absolute error of 0.173 and an R^2 of 0.966, clearly exceeding imputation methods (the errors around 0.25). The NMHC(GT) feature has an 88.4% missing rate, making imputation unreliable and introducing noise. Although direct deletion reduces the sample from 7,674 to 827 observations, it safeguards data integrity. This research shows that preserving data quality should take priority over quantity when the missing rates are exceptionally high, providing the practical guidance for robust preprocessing design in federated learning systems.

Keywords: Federated learning, air quality prediction, missing data imputation, data preprocessing.

1. Introduction

Air pollution, more to the point, the type of it, which is called ground level ozone, poses a very difficult challenge for the health of the world, not mentioning that it is also a problem to the environment. When making reliable forecasts of the quality of the air, then these forecasts can be used to support early warning mechanisms, and even help policy makers to make better decisions. The manner in which the monitoring stations are distributed in various geographical locations however, contributes to the fracturing of the data gathered into distinct and independent silos. Such a scenario renders it challenging to effectively employ conventional centralized machine learning methods.

There are various obstacles to centralized learning, such as privacy regulations, legal regulations, and cost of data transfer. Federated learning (FL) is a new distributed model that allows building models together without exposing raw data [1, 2]. With this design a client performs his or her task of optimizing a local model and transmits only the updates to the parameters of that model to a central aggregator [2, 3]. It is important to note that despite the fact that FL does not imply direct disclosure of original records, recent discoveries indicate that gradient exchanges could still expose confidential information [4]. This issue reveals the necessity to enhance data management processes within FL settings.

Recent advances in federated learning (FL) have primarily been concerned with the enhancement of algorithms, such as Non-IID data and reduction of communication costs [5, 6]. However, there is one important area that receives little attention: the impact of preprocessing data on FL outcomes. In regular centralized learning, model accuracy and generalization are evidently influenced by steps such as filling missing values and feature selection [7, 8]. Such as, Seu et al. investigated intelligent approaches to fill in the missing data [8], and Liu et al. provided an extensive overview of the methods to fill in missing data [9]. But how these concepts are transferred to FL environments where data are stored on a variety of types of clients, remains a question. This problem is significant to air quality

forecasting, as sensor networks tend to create incomplete records, outliers, and discontinuities across various monitoring locations.

It is an inquiry that conducts a systematic assessment of the impact of various preprocessing strategies that can be applied in the context of air quality prediction within the framework of federated learning. Based on UCI Air Quality data [10], the research examines four variants: direct deletion of incomplete records, which will be the control group; second is mean imputation, third is KNN imputation, and fourth is RF imputation. It is configured with five clients, and the performance is measured by MAE and RMSE. The findings confirm that the preprocessing method used is strongly affecting the model fidelity. Direct deletion produced a better result, having an MAE of 0.173 and an R2 of 0.966, and outperforming mean imputation (MAE of 0.249), KNN imputation (MAE of 0.253), and RF imputation (MAE of 0.250). Among the things to observe is that the NMHC(GT) feature had a rate of missing of 88.4 percent. Thus, although direct deletion will decrease the sample size of 7, 674 to 827 observations, it will still preserve the integrity of the data, and will not introduce the spurious noise that imputation methods tend to do.

2. Method

2.1. Data Preparation

2.1.1. Data Sources

This study uses the UCI Air Quality dataset [10], which records air monitoring data from an Italian location between March 2004 and April 2005. The original dataset has 9,471 entries spread across 15 attributes. After removing incomplete observations, it ends up with 7,674 valid instances. The prediction target is the ozone sensor output, labeled as PT08.S5 (O3), which is a continuous numeric variable and makes this a regression problem. Based on correlation analysis and domain knowledge, four predictors are included: CO(GT) for carbon monoxide, temperature (T), relative humidity (RH), and absolute humidity (AH). Their correlation coefficients with the target are 0.62, 0.45, -0.32, and 0.38 respectively.

2.1.2. Data Preprocessing

The NMHC(GT) feature had an 88.4% missing rate and was not used as a predictor. However, during initial cleaning, listwise deletion removed any record missing this feature. So for direct deletion, the sample dropped from 7,674 to 827 observations. Imputation methods filled those gaps and kept all records.

This study breaks data preprocessing into three phases: date and time conversion, missing value management, and feature standardization. It then compares four missing data approaches: direct deletion, mean substitution [8], KNN imputation with K set to 5[11], and random forest imputation [12]. The KNN technique finds the k closest records to an incomplete observation and uses a weighted average to fill the gap [11]. The random forest method predicts missing values by iteratively building ensemble tree models [12]. Finally, Z-score standardization was applied to remove the influence of different units [13], and the formula is given as follows.

$$x' = (x - \mu)/\sigma \tag{1}$$

The notation defines x' as the standardized value, x as the original measurement, μ as the feature mean, and σ as the feature standard deviation.

2.2. Federated Learning Prediction Models

2.2.1. Federated Learning Framework

Federated learning lets multiple parties train a model together, without directly sharing any raw data. The FedAvg algorithm is a standard example, and it uses a client-server setup for synchronous coordination. In each communication round, the server sends the global model to the clients. After getting that model, the clients do local training on their own datasets. When training finishes, the

clients send back their updated model parameters. The server then calculates a weighted average to create a new global model [1]. If the data distributions across clients differ a lot, this creates notable challenges for getting FedAvg to converge [14]. In this study, the authors simulate the federated learning environment by randomly spreading records across five clients, then combining the results after each training round, and finally repeating the whole process for ten cycles.

2.2.2. Neural Network Framework

This study uses a Multi-layer Perceptron, or MLP for short, as the main predictive model, and it runs inside the PyTorch environment. An MLP is a feedforward neural network that progressively pulls out features across its fully connected layers. This kind of design helps it capture the nonlinear relationships between the input variables and the target output. The network also uses ReLU activation functions to add nonlinearity, which lets the model pick up on complex patterns in the data. Table 1 shows the exact setup for the network architecture.

Table 1. Multi-layer Perceptron Network Architecture

Layer	Input Dimensions	Output Dimensions	Activation Function
Input Layer	4	64	-
Hidden Layer 1	64	64	ReLU
Hidden Layer 2	64	32	ReLU
Output Layer	32	1	Linear

The hyperparameters are set to a learning rate of 0.002, three rounds of local training, a batch size of 32, ten rounds of global communication, and a random seed of 42.

2.2.3. Evaluation Metrics

Three regression metrics are used to evaluate how well the model performs. The first one is MAE, which measures the average absolute deviation between the predicted values and the actual values. It treats all errors in an equal way and is not sensitive to outliers, so it can reflect the overall level of prediction bias. The formula for MAE is given as follows:

$$\mathrm{MAE}=\frac{1}{n}\sum_{i=1}^{n}|y_i-\hat{y}_i| \tag{2}$$

RMSE is calculated by first squaring the errors, then taking the average of those squared errors, and finally taking the square root. This metric assigns greater weight to larger errors and is more sensitive to them, so it ends up amplifying the model's prediction bias when extreme conditions occur. The formula for RMSE is given as follows:

$$\mathrm{RMSE}=\sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i-\hat{y}_i)^2} \tag{3}$$

The R^2 metric quantifies how much of the variability in the observed data can be accounted for by the model. In other words, it expresses the share of variance in the actual values that the predicted estimates manage to capture. This statistic runs along a scale from 0 to 1, where values that get close to 1 indicate a superior alignment between the model and the underlying data. The corresponding equation appears below.

$$R^2=1-\frac{\sum(y_i-\hat{y}_i)^2}{\sum(y_i-\bar{y})^2} \tag{4}$$

Here, n denotes the total number of samples, yi denotes the true value of the i-th sample, and yi denotes the predicted value of the i-th sample, and $\bar{y}=\frac{1}{n}\sum_{i=1}^{n}y_i$ is the mean of the true values.

3. Results and Discussion

3.1. Experimental Results

This study compares the performance of four preprocessing methods in a federated learning task for air quality prediction, using metrics such as MAE, RMSE, and R^2.

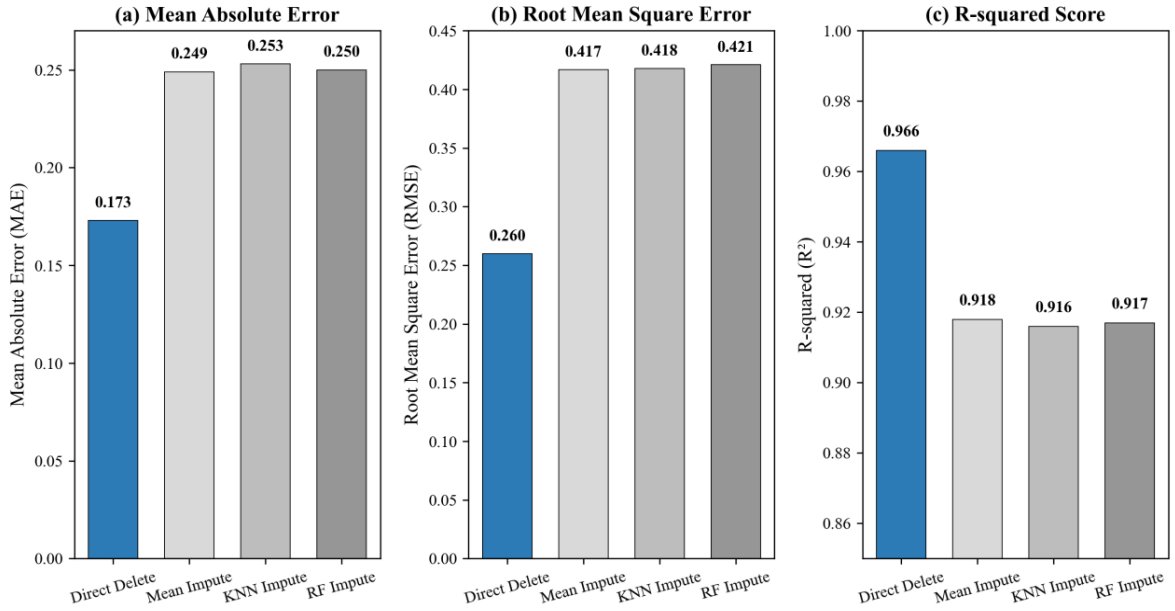

Fig. 1 Performance comparison of various models (Picture credit: Original)

Fig. 1 compares the performance of the four methods across three metrics. As shown, the direct deletion method performs best on all metrics. Its MAE is 0.173, which is significantly lower than that of mean replacement (0.249), KNN replacement (0.253), and RF replacement (0.250). The R^2value reaches 0.966, which is far higher than the 0.916–0.918 values achieved by the replacement methods.

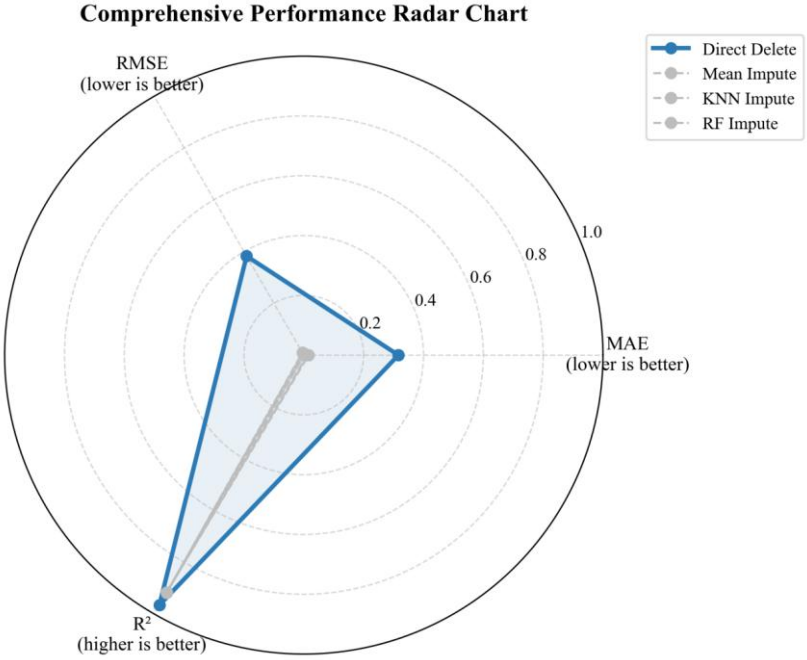

Fig. 2 Radar chart of model performance across multiple metrics (Picture credit: Original)

The radar chart in Fig. 2 provides a comprehensive comparison of the four methods across three dimensions. The direct deletion method demonstrates a clear advantage in all dimensions, with the largest polygon area, while the performance curves of the three filling methods overlap significantly, indicating similar performance.

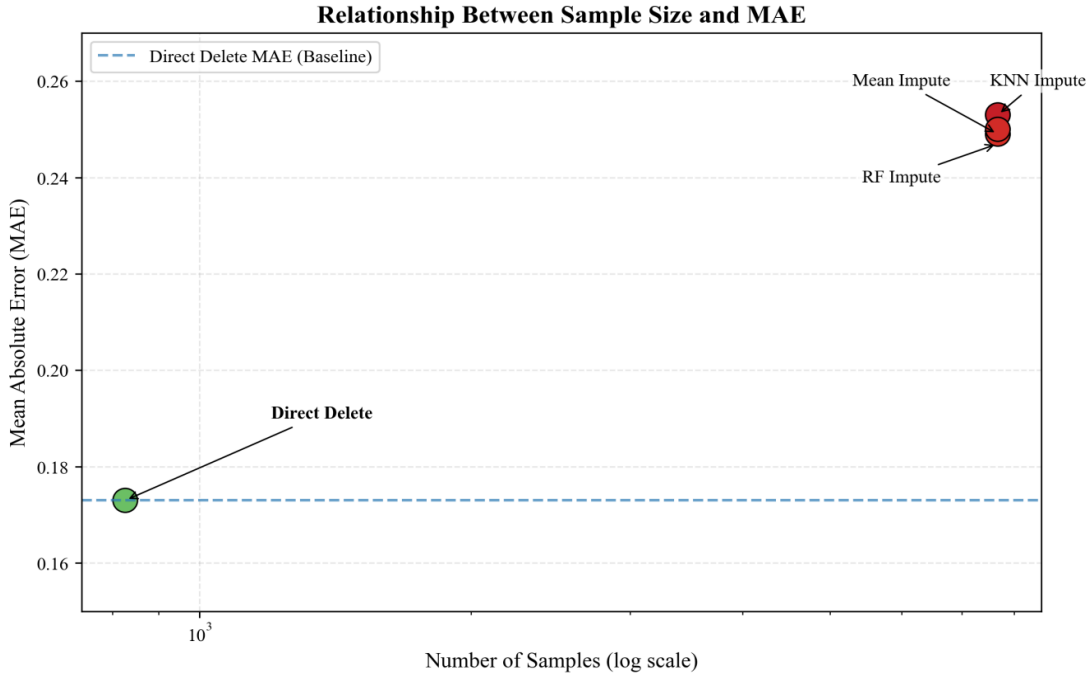

Fig. 3 Relationship between sample size and MAE (Picture credit: Original)

Fig. 3 illustrates the relationship between sample size and MAE. It is worth noting that while the direct deletion method uses the smallest sample size (827), it achieves the lowest MAE (0.173). In contrast, the three filling methods retain all 7,674 samples, but their MAE values are all around 0.25.

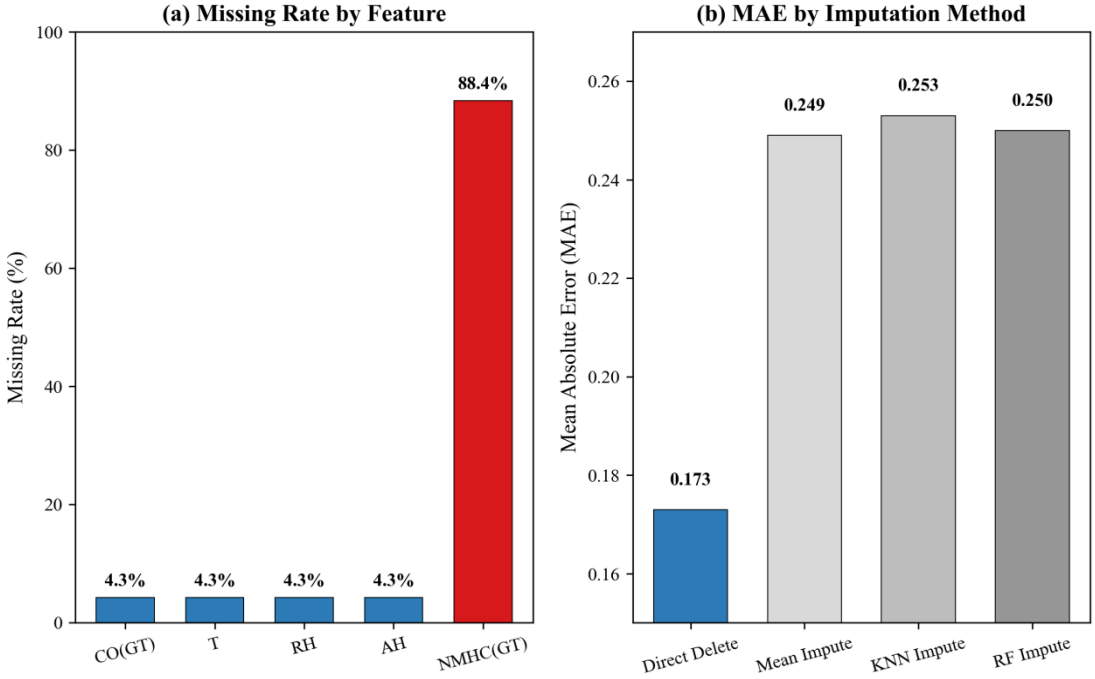

Fig. 4 Missing rate analysis of features and MAE comparison (Picture credit: Original)

Fig. 4 analyzes the relationship between the missing value rates of each feature and the performance of the imputation methods. The left panel of Fig. 4 shows that the missing value rate for the NMHC(GT) feature is as high as 88.4%, which is significantly higher than that of the other four features (approximately 4.3%). This is the primary reason for the failure of the imputation methods.

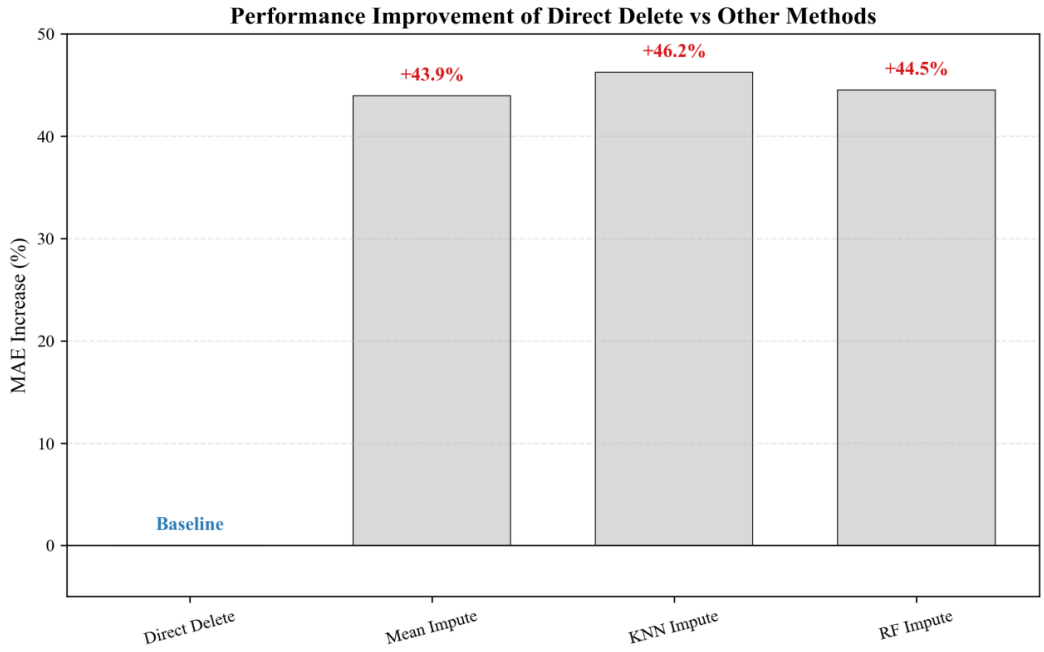

Fig. 5 MAE increase percentage of direct delete vs other imputation methods (Picture credit: Original)

Fig. 5 quantifies the performance improvement achieved by the direct deletion method: a 43.9% improvement over mean filling, a 46.2% improvement over KNN filling, and a 44.5% improvement over RF filling.

3.2. Analysis and Discussion

The reason why the direct deletion method performs best lies in data quality. The 88.4% missing rate in the NMHC(GT) feature means that any imputation method inevitably introduces noise. Mean imputation distorts the data distribution; KNN imputation distorts the distance metric under high missing rates; and RF imputation relies on incomplete features [7]. Although direct deletion reduces the sample size, it retains fully valid data points without introducing artificial noise.

In contrast to previous studies, Lee and Kang [15] note that when the missing value rate exceeds 70%, the performance of imputation methods typically declines significantly, to the point where they may even perform worse than simply deleting the missing values. The results of this study are highly consistent with this finding. Although some studies recommend RF or MICE as general-purpose imputation methods [7], these methods lose all their advantages under an extreme missing rate of 88.4%. This study further confirms that for features with extremely high missing rates, deletion should be prioritized over imputation.

The contribution of this study lies in providing empirical evidence for preprocessing strategies in environments with high missing values, cautioning researchers against blindly applying complex imputation algorithms, and offering direct practical guidance for the preprocessing of air quality datasets.

The limitations of this study include the use of a single dataset, the need for further validation of the conclusions, and the exclusion of more advanced imputation techniques such as MICE and deep learning methods.

4. Conclusion

This study addresses the question raised in the introduction: whether the role of data preprocessing in federated learning has been overlooked. Experimental results show that when the feature missing rate reaches as high as 88.4%, directly removing missing samples (MAE = 0.173, R^2=0.966) significantly outperforms mean imputation, KNN imputation, and random forest imputation. This

validates that, in scenarios with high missing rates, preserving the true data is more critical than increasing the sample size.

This study has certain limitations, as it relies on a single dataset; therefore, the conclusions need to be validated in a wider range of scenarios. Future work will focus on designing adaptive preprocessing strategies that account for missing data mechanisms, exploring robust models capable of handling extremely high missing rates, and providing more systematic guidance for data preprocessing in federated learning.

References

[1] McMahan B, Moore E, Ramage D, Hampson S, y Arcas BA. Communication-efficient learning of deep networks from decentralized data. In: Proc 20th Int Conf Artif Intell Stat (AISTATS); 2017. p. 1273–1282.

[2] Kairouz P, McMahan HB, Avent B, Bellet A, Bennis M, Bhagoji AN, et al. Advances and open problems in federated learning. Found Trends Mach Learn. 2021; 14(1–2): 1–210.

[3] Yang Q, Liu Y, Chen T, Tong Y. Federated machine learning: Concept and applications. ACM Trans Intell Syst Technol. 2019; 10(2): 1–19.

[4] Zhu L, Liu Z, Han S. Deep leakage from gradients. In: Proc Adv Neural Inf Process Syst (NeurIPS); 2019. p. 14774–14784.

[5] Li T, Sahu AK, Talwalkar A, Smith V. Federated learning: Challenges, methods, and future directions. IEEE Signal Process Mag. 2020; 37(3): 50–60.

[6] Li T, Sahu AK, Zaheer M, Sanjabi M, Talwalkar A, Smith V. FedProx: Federated learning with proximal term. In: Proc Mach Learn Syst (MLSys); 2020. p. 429–450.

[7] 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(1): 1–37.

[8] Seu K, Kang MS, Lee H. An intelligent missing data imputation techniques: A review. Int J Informatics Vis. 2022; 6(1–2): 278–283.

[9] Liu M, Li S, Yuan H, Ong MEH, Ning Y, Xie F, et al. Handling missing values in healthcare data: A systematic review of deep learning-based imputation techniques. Artif Intell Med. 2023; 142: 102587.

[10] De Vito FS, Massera E, Piga M, Martinotto L, Di Francia G. Air quality data set. UCI Mach Learn Repository. 2008. Available from: https://archive.ics.uci.edu/ml/datasets/Air+Quality

[11] Khan SI, Hoque ASML. SICE: An improved missing data imputation technique. J Big Data. 2020; 7(1): 1–21.

[12] Ali SS, Shaikh F, Shah SA. Missing data imputation using random forests and its application in healthcare. In: Proc Int Conf Comput Intell (ICCI); 2020. p. 152–157.

[13] Yıldız A, Er ME, Bursalı A, Çolakoğlu T, Erkuş EC. The effects of data standardization and normalization techniques in click through rate prediction. In: Proc Int Conf Adv Technol; 2023. p. 200–203.

[14] Mora A, Bujari A, Bellavista P. Enhancing generalization in federated learning with heterogeneous data: A comparative literature review. Future Gener Comput Syst. 2024; 157: 1–15.

[15] Lee KJ, Kang JB. The proportion of missing data should not be used to guide decisions on multiple imputation. BMC Med Res Methodol. 2019; 19: 159.