



# Imputation Method Using Local Linear Regression Based on Bidirectional $k$ -nearest-components

Yonggeol Lee\* , Member, KIICE

Division of Software Convergence, Hanshin University, Osan 18101, Republic of Korea

## Abstract

This paper proposes an imputation method using a bidirectional  $k$ -nearest components search based local linear regression method. The bidirectional  $k$ -nearest-components search method selects components in the dynamic range from the missing points. Unlike the existing methods, which use a fixed-size window, the proposed method can flexibly select adjacent components in an imputation problem. The weight values assigned to the components around the missing points are calculated using local linear regression. The local linear regression method is free from the rank problem in a matrix of dependent variables. In addition, it can calculate the weight values that reflect the data flow in a specific environment, such as a blackout. The original missing values were estimated from a linear combination of the components and their weights. Finally, the estimated value imputes the missing values. In the experimental results, the proposed method outperformed the existing methods when the error between the original data and imputation data was measured using MAE and RMSE.

**Index Terms:** Data Imputation, Missing Data, Bidirectional  $knc$ , Local Linear Regression, Blackout

## I. INTRODUCTION

Missing problems caused by technical issues, such as errors or breakdowns, appear at various stages of the data collection process [1,2,3]. In general, the value of the point or block where the missing occurs is filled with "NULL." Missing values can lead to biased results and affect the performance of machine learning algorithms [1,3,4]. In particular, "blackouts" are extreme missing scenarios, in which all the sensors are quiet simultaneously, causing widespread and aligned missing blocks [5]. Until recently, few algorithms have imputed missing blocks with high accuracy in blackouts [5].

Various methods have been proposed for addressing this problem [6-10]. Traditional methods impute missing values using neighboring data components. The last observation carried forward (*locf*) [6], next observation carried backward

(*nocb*), and *nearest* methods impute the missing values from a missing point to its nearest component. These methods are fast and have high imputation performance; however, their performance degrades dramatically when missing scenarios, such as a blackout, occur [7].

In addition, the *mean* method [8,9] estimates the missing values by linearly combining the surrounding components and weights based on a window within the given data. The  $k$ -nearest neighbor (*knn*) method [10] estimates the imputation values from the neighboring data closest to the given data in an entire dataset. Despite these methods showing high imputation performance, they are limited when the missing rate increases. The *mean* method cannot estimate a missing value when all the neighboring values in the window are missing. The *knn* method does not operate normally when a missing value occurs commonly at the same point in the entire dataset. The *mean* method cannot estimate a miss-


Received 2 October 2022, Revised 26 October 2022, Accepted 30 October 2022

\*Corresponding Author Yonggeol Lee (E-mail: [pattern@hs.ac.kr](mailto:pattern@hs.ac.kr), Tel: +82-31-379-0656

Division of Software Convergence, Hanshin University, Osan 18101, Republic of Korea

Open Access <https://doi.org/10.56977/jicce.2023.21.1.62>

print ISSN: 2234-8255 online ISSN: 2234-8883

 This is an Open Access article distributed under the terms of the Creative Commons Attribution Non-Commercial License (<http://creativecommons.org/licenses/by-nc/3.0/>) which permits unrestricted non-commercial use, distribution, and reproduction in any medium, provided the original work is properly cited.

Copyright © The Korea Institute of Information and Communication Engineering

ing value when all the values in the window are missing, and the  $knn$  method does not operate normally when a missing value occurs in common with the same data point in all the datasets.

This paper proposes an imputation method for an environment with a high missing rate, such as a blackout. The bidirectional  $k$ -nearest-components search method selects the components in the dynamic range from the missing points. The weight values assigned to the components around the missing points are calculated using local linear regression [11]. The original values of the missing positions are estimated from a linear combination of the neighboring components and their weights. Finally, the estimated value imputes the missing values. In the experimental results, the proposed method showed superior performance compared to that of the existing methods when the error between the original data and imputation data was measured using the mean absolute error (MAE) and root mean square error (RMSE).

The remainder of this paper is organized as follows. Section II describes the proposed method for imputing missing data. This explains the bidirectional  $k$ -nearest-components search method and missing data imputation using local linear regression. The experimental results of data imputation are described in Section III. Discussion and conclusions are presented in Section IV.

## II. PROPOSED METHODS

This paper proposes an imputation method using a bidirectional  $k$ -nearest-components-search-based local linear regression method. The proposed method estimates missing values from the adjacent components of the missing points. The overall process is as follows:

1. The occurrence of the missing is checked for each data point.
2. When a missing point occurs at a given data point,  $k$ -components (normally measured) and corresponding location information are searched in both directions based on the point.
3. The local linear regression method is applied to estimate the imputation values for the missing values. The regression analysis determines the optimal parameters (weights), for which the residual is minimized.
4. Finally, a weight is applied to the input of the given data ( $k$ -components) to estimate the missing values. The estimated values impute the missing values.

The overall flow of the proposed method is shown in Fig. 1.

### A. Bidirectional $k$ -nearest-components

In the imputation problem, interpolation methods, such as the *mean* method, estimate missing values from components

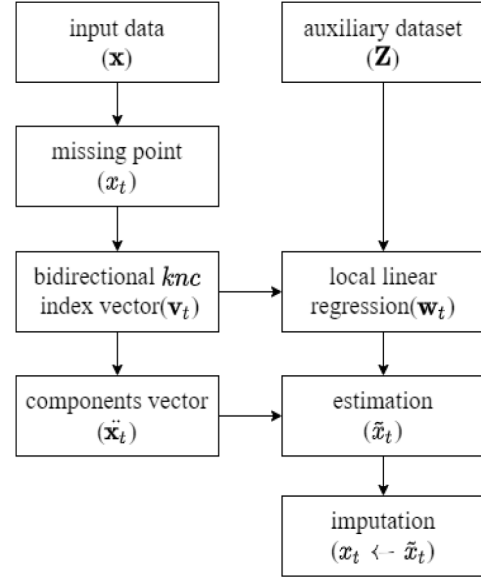


Fig. 1. Overall procedure of the proposed method.

within a symmetrically fixed range called a “window” around the missing point. These methods exhibit relatively high imputation performance in an ideal situation where no missing data are available. However, these methods are limited when the missing rate ( $l$ ) is high.

We considered strongly correlated time-series data. In Fig 2, the missing value,  $x_t$  can be assumed 10. In Fig. 2(a), the *mean* method estimates the missing values as 10 by applying a single weight of 0.25 to all the components. The *mean* method works well in an ideal situation without missing values ( $l = 0\%$ ). However, as the missing rate within the window increases, the imputation performance gradually decreases. Fig. 2 (b) shows a significant difference between the original and estimated values when the missing rate reaches 50% within the window. As shown in Fig. 2 (c), missing values cannot be imputed when  $l$  becomes 100% extremely. Therefore, the value of  $x_t$  is still unchanged. Consequently, the imputation performance depends on the missing rate within a fixed window size.

Therefore, in an imputation problem, the selection of the adjacent components must be flexible. This paper proposes a bidirectional  $k$ -nearest-components ( $bknc$ ) search method to select components from the missing point. The  $bknc$  search method finds the left and right non-missing  $k$ -components centered on the missing point. Accordingly, the data values and respective index information of  $2k$ -components are extracted. Because  $bknc$  searches for dynamic ranges, the window size is symmetric or asymmetric depending on the missing point.

The function  $\mathcal{L}(x_t)$  for determining whether the input data  $\mathbf{x} \in \mathcal{R}^d$  is missing for each data point ( $x_t$ ) is defined as:

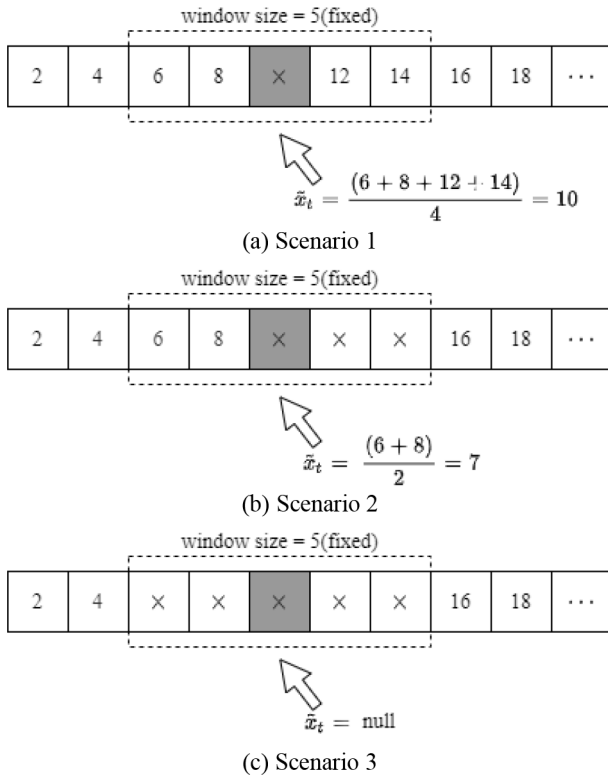


Fig. 2. Missing scenarios at fixed window size.

$$\mathcal{L}(x_t) = \begin{cases} 0, & \text{missing is occurred} \\ 1, & \text{otherwise.} \end{cases} \quad (1)$$

Based on the missing point  $t$ , *bknc* searches each normally measured  $k$ -component in both the directions and extracts the corresponding index vector  $\mathbf{v}_t \in \mathfrak{R}^{2k}$ .

In Fig. 2(c),  $\mathbf{v}_t$  is composed of  $[t - 5, t - 3, t + 3, t + 4]$  when  $k = 2$  is applied. Therefore, the overall window size becomes 10, and the start and end indices become  $\min(\mathbf{v}_t)$  and  $\max(\mathbf{v}_t)$ , respectively.

### B. Weight Assignment and Imputation

In general, weights are required to be applied to neighboring components for imputation. Traditional methods, such as *locf* (left side), *nocb* (right side), and *nearest* (both sides), assign 100% weight to the component nearest to the missing point. Other methods apply individual weights to all the missing values within a fixed-size window. The method of calculating the weight is to assign the same weight ( $\nabla w = 1 / \sum_{i=-k}^k \mathcal{L}(x_{t+i})$ ) to all the positions or to assign the weight inversely proportional to the distance ( $w = 1/\text{distance}$ ). Traditionally, these methods reduce the average error between the original and imputed data; however, it is difficult to reflect the data flow in a specific environment, such as a blackout.

In this paper, the weights were calculated using local linear regression. Regression determines the optimal weights (parameters) for estimating the dependent variable from the independent variables [12]. Accordingly, it is possible to impute a value that reflects the data flow by assigning an appropriate weight to the surrounding values from the position where the missing data occur. To calculate weights from local linear regression, it is necessary to construct an independent variable matrix and dependent variable vector from the auxiliary set  $\mathbf{Z} \in \mathfrak{R}^{m \times d}$ . Higher performance can be expected when an auxiliary set without missing values is assigned.

$$\mathbf{z}_t = \ddot{\mathbf{Z}}_t \cdot \mathbf{w}_t$$

$$\begin{bmatrix} z_t^1 \\ z_t^2 \\ \vdots \\ z_t^m \end{bmatrix} = \begin{bmatrix} z_{v_1}^1 & \dots & z_{v_{2k}}^1 \\ z_{v_1}^2 & \dots & z_{v_{2k}}^2 \\ \vdots & \ddots & \vdots \\ z_{v_1}^m & \dots & z_{v_{2k}}^m \end{bmatrix} \begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ w_{2k} \end{bmatrix}. \quad (2)$$

In Eq. (2), the independent variable matrix  $\ddot{\mathbf{Z}}_t \in \mathfrak{R}^{m \times 2k}$  is composed of column vectors corresponding to  $\mathbf{v}_t$  in the auxiliary dataset  $\mathbf{Z} \in \mathfrak{R}^{m \times d}$ ; the dependent variable vector  $\mathbf{z}_t \in \mathfrak{R}^m$  becomes the column vector at point  $t$  in  $\mathbf{z}$ . Note that all the values are normally measured. Finally, the weight vector  $\mathbf{w}_t \in \mathfrak{R}^{2k}$  is defined as:

$$\mathbf{w}_t = \text{inv}(\mathbf{Z}_t^T \mathbf{Z}_t) (\mathbf{Z}_t^T \mathbf{z}_t) \quad (3)$$

In Eq. (3),  $(\mathbf{Z}_t^T \mathbf{Z}_t)$  is invertible ( $d \gg m > k$ ) because the matrix  $\ddot{\mathbf{Z}}_t$  is full column rank.

The final step is the imputation phase. From the index vector  $\mathbf{v}_t$  extracted from the *bknc*, components vector  $\ddot{\mathbf{x}}_t \in \mathfrak{R}^{2k}$  of  $\mathbf{x}_t$  is constructed. From the linear combination of  $\ddot{\mathbf{x}}_t$  and  $\mathbf{w}_t$ , the missing value  $\hat{x}_t$  can be estimated as:

$$\hat{x}_t = \ddot{\mathbf{x}}_t \cdot \mathbf{w}_t \quad (4)$$

Finally, the missing value  $x_t$  is imputed by  $\hat{x}_t$ .

### III. RESULTS

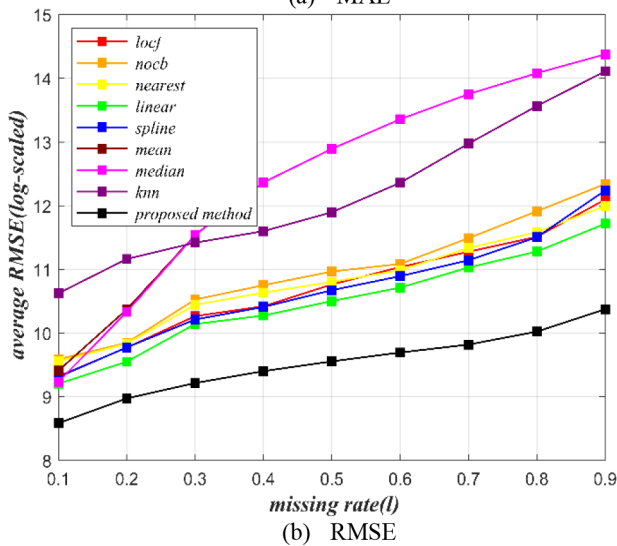
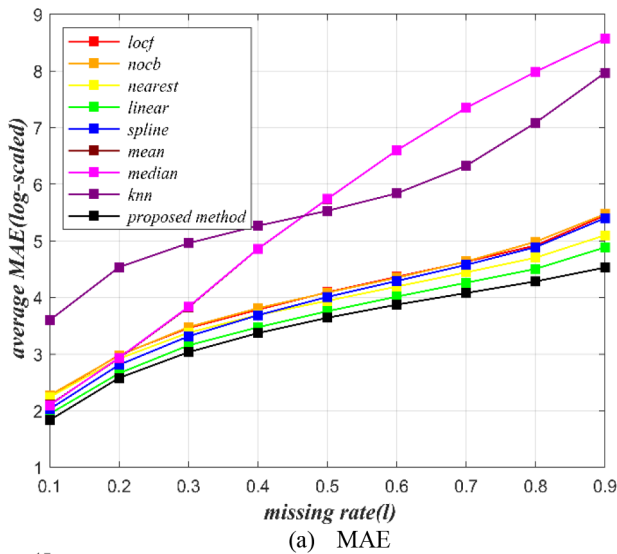
In the experiment, electronic nose (E-nose) data [13] were used to confirm the imputation performance of the proposed method. The E-nose consisted of eight gases measured using 16 sensor arrays. Each sensor was recorded at a sampling rate of 10 Hz for 200s. Consequently, the sensor array was stored in the form of a  $2,000 \times 16$  matrix, which was converted into a  $1 \times 32,000$ -dimensional vector for use in the experiment. Missing data were generated by randomly assigning missing values (“NULL”) to points in the 32,000 dimensions according to the missing rate ( $l = 0.1 \sim 0.9$ ). The experiment excluded  $l = 1$ , where all data points are “NULL”.

The imputation performance of each method was evaluated by measuring the MAE and RMSE of the imputed and original data. MAE and RMSE are defined as:

$$MAE = \frac{1}{d} \sum_{t=1}^d |x_t - \tilde{x}_t|$$

$$RMSE = \sqrt{\frac{1}{d} \sum_{t=1}^d (x_t - \tilde{x}_t)^2} \quad (5)$$

In addition, the proposed method was compared with the experimental results of existing methods, such as *locf*, *nocb*, *nearest*, *linear*, *spline*, *mean*, *median*, and *knn*. In the initial parameter-setting stage, the window size was set to five for the mean and median methods using the windows. In the proposed method, a dynamic window size was created for



**Fig. 3.** Comparison of error between imputed and original data according to missing rate (0.1~0.9) (a) MAE (b) RMSE

each data point by setting parameter  $k$  of *bknc* to 2.

Figs. 3(a) and (b) show the measurement results of MAE and RMSE according to the missing rate. All the numerical values were adjusted to a logarithmic scale. From the experimental results, the proposed method exhibits the best performance for all the missing rates. Interpolation-based imputation methods (*linear* and *spline*) [14] and single-component-based methods (*locf*, *nocb*, and *nearest*) performed well, in that order. In the window-based *mean* and *median* methods, the performance decreased sharply from a missing rate of 0.3. However, *knn* showed a robust imputation performance at a higher missing rate than at lower missing rate.

As listed in Table 1, when the missing rate is 0.1, the proposed method shows an MAE lower than 0.0001~0.0030 compared to that of the existing methods. There were differences between 0.0005~0.0273 ( $l = 0.5$ ), and 0.0040~0.5166 ( $l = 0.9$ ), even when the missing rate increased. In Table 2, the mean RMSE values according to the missing rate were 0.4601~3.5811 ( $l = 0.1$ ), 2.2301~38.3308 ( $l = 0.5$ ), and 9.0724~172.1899 ( $l = 0.9$ ), respectively. As a result, the proposed method outperformed other methods for both the indicators.

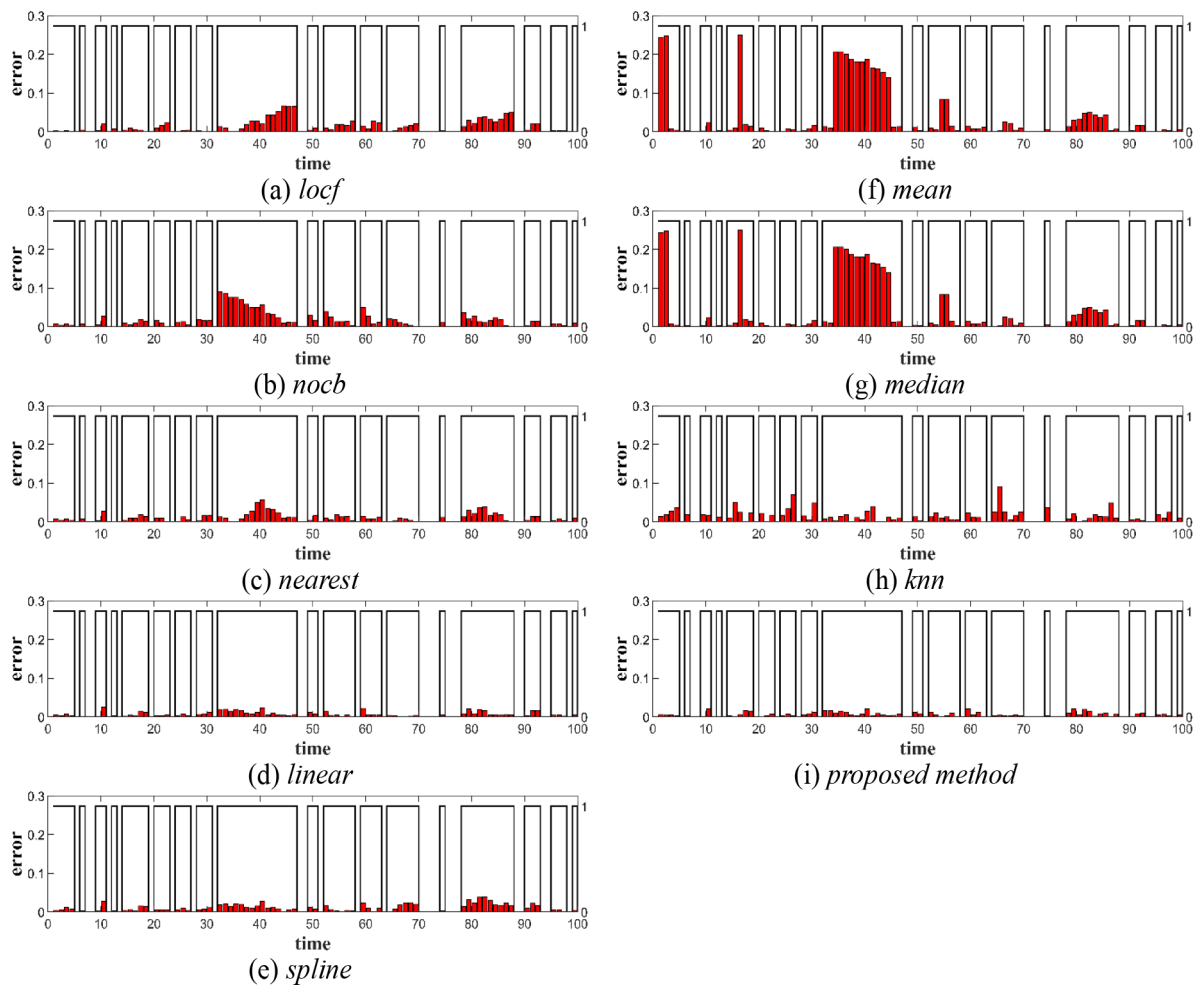
Fig. 4 shows the results of the MAE measurements between the imputed and original data for each data point of the first sensor during the initial 10s interval when  $l = 0.7$ . Existing methods do not operate normally when a blackout occurs, as the missing rate increases. As shown in Table 1, the proposed method exhibits a high imputation performance. However, as mentioned earlier, the *mean* and *median* methods using a fixed-size window failed to perform imputation when the missing block size was larger than the window size, resulting in a significant error value. The *knn* method had a large error when a missing value occurred in common with the same data point in all the datasets.

## IV. DISCUSSION AND CONCLUSIONS

Missing data is an unavoidable problem in the real world. Various methods have been proposed to solve this problem; however, they have limitations in extreme situations, such as blackouts.

In this paper, we propose a robust imputation method for blackouts. The proposed method has several advantages.

The proposed method can secure the number of variables required for modeling regardless of the missing rate. In addition, the value can be estimated by reflecting the flow of data using the regression analysis. Furthermore, modeling is possible without the problem of lack of rank in the matrix of dependent variables using the local linear regression analysis. In the experimental results, MAE and RMSE were measured to verify the imputation performance. The proposed method was superior to the existing methods and performed robustly even when a blackout occurred, owing to an increase



**Fig. 4.** MAE results between imputed and original data at  $l = 0.7$ .

in the missing rate.

However, the proposed method is limited because a separate clean dataset is required. Similar to the *knn* method, the imputation performance is significantly lowered if the neighbor is used only within the entire dataset. In addition, the determination of the number of components ( $k$ ) was passive.

Despite some limitations, the proposed method showed high imputation performance, even with an increase in the missing rate. Consequently, the proposed method can address the missing problems in the real world. In the future, machine learning-based methods may be considered when estimating missing values.

**Table 1.** Mean absolute error between the imputed and original data

Method	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
<i>locf</i>	0.0010	0.0020	0.0032	0.0044	0.0060	0.0079	0.0103	0.0137	0.0232
<i>nocb</i>	0.0010	0.0020	0.0032	0.0045	0.0060	0.0077	0.0103	0.0146	0.0238
<i>nearest</i>	0.0009	0.0019	0.0029	0.0040	0.0052	0.0066	0.0085	0.0110	0.0164
<i>linear</i>	0.0007	0.0014	0.0023	0.0032	0.0043	0.0055	0.0071	0.0090	0.0132
<i>spline</i>	0.0008	0.0017	0.0028	0.0040	0.0055	0.0073	0.0097	0.0133	0.0222
<i>mean</i>	0.0008	0.0019	0.0046	0.0129	0.0311	0.0733	0.1551	0.2938	0.5258
<i>median</i>	0.0008	0.0019	0.0046	0.0129	0.0311	0.0733	0.1551	0.2938	0.5258
<i>knn</i>	0.0037	0.0093	0.0143	0.0194	0.0252	0.0343	0.0559	0.1194	0.2885
<b>proposed method</b>	<b>0.0006</b>	<b>0.0013</b>	<b>0.0021</b>	<b>0.0029</b>	<b>0.0038</b>	<b>0.0048</b>	<b>0.0059</b>	<b>0.0073</b>	<b>0.0093</b>

**Table 2.** Root mean square error between the imputed and original data

Method	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
<i>locf</i>	1.1189	1.7554	2.8781	3.3571	4.7288	6.2128	7.9067	9.9958	17.9357
<i>nocb</i>	1.4619	1.8928	3.7340	4.6760	5.7957	6.5312	9.7803	14.9252	22.9156
<i>nearest</i>	1.4122	1.8694	3.4337	4.1596	4.9267	5.9168	8.3867	10.7663	16.2364
<i>linear</i>	0.9961	1.4074	2.5418	2.9053	3.6446	4.5023	6.1787	7.9292	12.2796
<i>spline</i>	1.1071	1.7618	2.7279	3.3222	4.3230	5.3873	6.9168	9.9372	20.6975
<i>mean</i>	1.2222	3.1959	10.2908	23.4784	39.7454	63.2239	93.8461	130.4195	175.3971
<i>median</i>	1.0255	3.0754	10.2596	23.4461	39.7299	63.2177	93.8423	130.4182	175.3967
<i>knn</i>	4.1171	7.0537	9.1154	10.9186	14.6979	23.3330	43.1346	77.8462	134.4522
<i>proposed method</i>	<b>0.5360</b>	<b>0.7906</b>	<b>1.0068</b>	<b>1.2135</b>	<b>1.4145</b>	<b>1.6254</b>	<b>1.8412</b>	<b>2.2565</b>	<b>3.2072</b>

## ACKNOWLEDGMENTS

This work was supported by Hanshin University Research Grant.

## REFERENCES

- [ 1 ] P. Bansal, P. Deshpande and S. Sarawagi, "Missing value imputation on multidimensional time series," *arXiv preprint arXiv:2103.01600*, Mar. 2021. DOI: 10.48550/arXiv.2103.01600.
- [ 2 ] Y. Lee, and S. I. Choi, "Data restoration by linear estimation of the principal components from lossy data," *IEEE Access*, vol. 8, pp. 172244-172251, 2020. DOI: 10.1109/ACCESS.2020.3024809.
- [ 3 ] T. Aittokallio, "Dealing with missing values in large-scale studies: microarray data imputation and beyond," *Briefings in bioinformatics*, vol. 11, no. 2, pp. 253-264, Mar. 2010. DOI: 10.1093/bib/bbp059.
- [ 4 ] S. Zhang, J. Zhang, X. Zhu, Y. Qin, and C. Zhang, "Missing value imputation based on data clustering," *Transactions on computational science I*, vol. 4750, pp. 128-138, 2008. DOI: 10.1007/978-3-540-79299-4\_7.
- [ 5 ] M. Khayati, A. Lerner, Z. Tymchenko, and P. Cudré-Mauroux, "Mind the gap: an experimental evaluation of imputation of missing values techniques in time series," in *Proceedings of the VLDB Endowment*, vol. 13, no. 5, pp. 768-782, 2020. DOI: 10.14778/3377369.3377383.
- [ 6 ] J. Shao and B. Zhong, "Last observation carry forward and last observation analysis," *Statistics in medicine*. Vol. 22, no. 15, pp. 2429-2441, Aug. 2003. DOI: 10.1002/sim.1519.
- [ 7 ] A. R. T. Donders, G. J. Van Der Heijden, T. Stijnen, and K. G. Moons, "A gentle introduction to the imputation of missing values," *Journal of clinical epidemiology*, vol. 59, no. 10, pp. 1087-1091, Oct. 2006. DOI: 10.1016/j.jclinepi.2006.01.014.
- [ 8 ] D. C. Howell, "The treatment of missing data," *The Sage handbook of social science methodology*, pp. 208, 2007.
- [ 9 ] G. Kalton and D. Kasprzyk, "Imputing for missing survey responses," *Proceedings of the section on survey research methods, American Statistical Association*, vol. 22, p. 31, American Statistical Association, Cincinnati, 1982.
- [ 10 ] J. K. Dixon, "Pattern recognition with partly missing data," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 9, no. 10, pp. 617-621, 1979. DOI: 10.1109/TSMC.1979.4310090.
- [ 11 ] J. Fan, "Local linear regression smoothers and their minimax efficiencies," *The annals of Statistics*, vol. 21, no. 1, pp. 196-216, Mar. 1993. [Online] Available: <https://www.jstor.org/stable/3035587>.
- [ 12 ] L. Ayalew and H. Yamagishi, "The application of GIS-based logistic regression for landslide susceptibility mapping in the Kakuda-Yahiko Mountains, Central Japan," *Geomorphology*, vol. 65, no. 1-2, pp. 15-31, Feb. 2005. DOI: 10.1016/j.geomorph.2004.06.010.
- [ 13 ] S. I. Choi, G. M. Jeong, and C. Kim, "Classification of odorants in the vapor phase using composite features for a portable e-nose system," *Sensors*, vol. 12, no. 12, pp. 16182-16193, 2012. DOI: 10.3390/s121216182.
- [ 14 ] D. Kahaner, C. Moler, and S. Nash, "Numerical methods and software," Prentice-Hall, 1989.



### Yonggeol Lee

received his B.S. degree in Applied Computer Engineering, and received M.S. and Ph. D. degrees in Computer science and Engineering degrees from Dankook University in 2012 and 2019, respectively. He worked at the Police Science Institute in Korean National Police University as a researcher from 2017 to 2021. He is currently an assistant professor with the Division of Software Convergence, Hanshin University. His research interests include image processing, computer vision, machine learning and pattern recognition.