Regular Paper

# Recommendation of Imputing Value for Sensor Data based on Programming by Example

Hiroko Nagashima[1,a) ] Yuka Kato[1,b)]

Received: May 8, 2019, Accepted: November 7, 2019

***Abstract:*** Large volumes of data are typically used during analyses. Data preprocessing, which involves detecting outliers, handling missing data, data formatting, integration, and normalization, is essential for achieving accurate results. Many tools and methods are available for reducing preprocessing time. However, most analysts face difficulties when using them. This paper proposes a method for handling outliers and missing data, called Automated PRE-Processing for Sensor Data (APREP-S). For reducing analysis resources, we combine programming by example and machine learning via Bayesian inference, inputting human knowledge to APREP-S as an example and calculating a proper proportion by machine learning via Bayesian inference. We also define k-Shape as the calculation of the rate of similarity of time-series data. In evaluation, we use sensor data of temperature and humidity and compare the sum of the square of the errors of four methods, between original data and outputs of each methods, (1) APREP-S, (2) mean of the entire data, (3) mean of the around-the-target imputation data, and (4) spline interpolation. It is verified that APREP-S is a more suitable method for humidity data than temperature data. preprocessing method. we consider the reason is that humidity data have more changing points.

***Keywords:*** pre-processing, sensor data, PBE (program by example), IoT, Bayesian inference

## 1. Introduction

In the field of information technology, it has recently become possible to analyze integrated data obtained from sensors or wearable devices, besides utilizing data on existing systems. Hence, large amounts of various data, including customer behavioral patterns in a shop, autonomous motion of robots, and fault detection during credit card use, can now be analyzed from multiple perspectives. However, these data often include outliers and missing data, inconsistencies in units and device specifications, or ambiguities within the data [1]. The data analysis flow involves "considering the aim of analysis", "transformation (preprocessing)", "creating the model", and "documenting for sharing knowledge", called data mining. The data mining flow in sensor data is shown in **Fig. 1**. In particular, data acquired from certain networks, including sensor data, require preprocessing owing to the noise and missing data, which inevitably occur because of being occasionally delayed or not being received by the load transfer. The examples of sensor data analysis include human vital data, smarthomes, and industrial prediction maintenance. These sensor analysis systems use 1) wireless networks, 2) sensors having a battery, and 3) time-series data. Therefore, the sensor data have more outliers and missing data than general time-series data because of collection through the network and use of a battery. It is necessary to check for outliers and missing data and to modify them as required. These processes, termed preprocessing, use 80% of the resources, even for an ordinary analysis system [2].

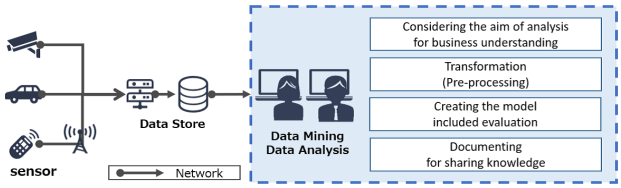


**Fig. 1** Overview of sensor data analysis.

An example of preprocessing is shown in **Fig. 2**. Two types of data are available from the sensors or dataset: weather data and person location data. The data include outliers and missing data; there are differences in data formats besides inconsistencies in units and interval time due to device specifications. Therefore, it is necessary to obtain a uniform format for integration, and depending on the aim of the analysis, outliers and missing data should be imputed. The preprocessing procedure for joining temperature data and person location data is as follows:

- Handling outliers or missing data: remove the data where $id = -1$ for weather data and calculate the mean of the *temperature* column if the data is NULL for imputation.
- Transformation of the time-series data: transforming Unix-time data format to "yyyy/MM/dd HH:mm", such as "2018-08-10 12:20".
- Discretization: make the interval of measurement time between two tables uniform for joining, which is the same as creating a join key in this case.

We previously proposed a data mining framework, called "APREP-DM (Automated PRE-Processing for Data Mining) [3]", which includes automated preprocessing to reduce tasks of analysts. The preprocessing in APREP-DM comprises

[1] Tokyo Woman's Christian University, Suginami, Tokyo 167–8585, Japan
[a)] h18m001@cis.twcu.ac.jp
[b)] yuka@lab.twcu.ac.jp

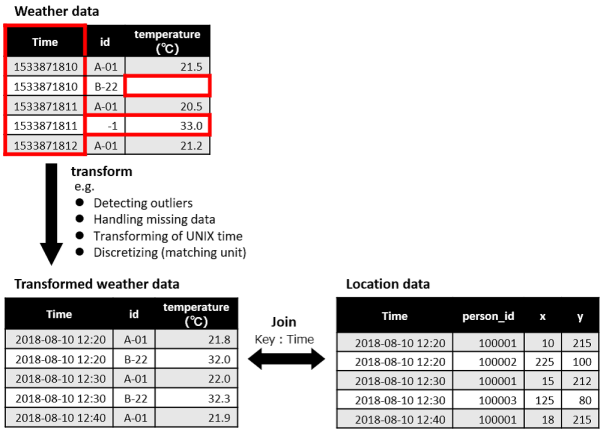


**Fig. 2** Example of preprocessing.

"common processes for data cleaning" and "other processes for data cleaning". Common processes for data cleaning can run automatically, e.g., detecting outliers or handling missing data. On the other hand, other processes for data cleaning need to find a suitable model through trial-and-error for obtaining the analysis result, that is, they cannot run automatically. In this paper, we consider outliers and missing data in "the common processes for data cleaning" in APREP-DM as the targets. We propose an analysis method that can reduce the required number of tasks via integration, "business understanding", which incorporates human knowledge and machine learning based on the programming by example (PBE) approach. For imputation of outliers and missing data, the analyst defines a rule, e.g., single imputation or multi-imputation. However, single imputation needs to define just one rule, and multi-imputation takes considerable time. Thus, we propose a method, called Automated Pre-Processing for Sensor data (APREP-S)," which integrates human knowledge and machine learning using the PBE approach. The specifics of single imputation and multi-imputation are described in Section 2.2, and those of PBE are described in Section 2.3. In this paper, we do not refer to the method of detecting outliers.

The following are the contributions of this paper.

- Proposing an automatic imputation method for outliers and missing data based on machine learning integrated with human knowledge using a PBE approach to reduce the number of analysis resources.
- To verify the effectiveness and suitable data of APREP-S using data that have outliers and missing data by comparing the accuracy of imputation with three existing imputation methods.

The imputation method determines how to calculate inputting values instead of outliers and missing data.

Section 2 presents a brief overview of the relevant literature. Section 3 shows the workflow of tasks related to the analysts and APREP-S. In Section 4, the features and details of APREP-S are described. Section 5 evaluates APREP-S, including the results and discussion of the evaluation. A summary of the main conclusions is presented in Section 6.

## 2. Related Work

The imputation method for outliers and missing data can be classified into two categories: "manual processing" and "automated processing". The "PBE approach" is a method for integrating the knowledge of the analyst into automated processing. These approaches are specifically described in the following sections.

### 2.1 Manual Processing

"Manual processing" is a method in which the analyst defines and develops the preprocessing processes on their own and checks the data profile to determine whether there are outliers, missing data, or inconsistencies in the format or spelling. To reduce the number of tasks that must be performed, several tools are available, e.g., OpenRefine [4] and Trifacta Wrangler [5]. These tools can assist analysts in sorting, aggregating, and detecting data that need to be transformed on the GUI. Moreover, analysts can iterate the process automatically if the flow is the same, using the record function of the process logs.However, analysts need to maintain on their own when the flow changes. In addition, they must consider the data handling rule of imputation and removal. That is, the analysts must select the correct operations and parameters for the data transformation task. Hence, data transformation tools are difficult to use for people who have no experience with them or programming skills [6]. Thus, we need a method using which the analysts can select a correct operation that requires fewer and easier tasks than manual processing.

### 2.2 Automated Processing

"Automated processing" is a method that imputes data without manual processing. It includes complete-case analysis (list-wise deletion), single imputation, and multiple imputation for MCAR. Missing data are often categorized into the following three types: missing completely at random (MCAR), missing at random (MAR), and missing not at random (MNAR) [7], [8]. As the missing data occur completely randomly in a sensor network, we consider MCAR data in this paper. The complete-case analysis (list-wise deletion) is a method of removing missing data. However, it might discard valuable data and weaken the statistical power because of a reduction in sample size. Single imputation is based on the same one rule, e.g., "inputting the mean of an entire column" or "inputting median of an entire column". However, the result is potentially biased and the amount of error difference increases when the range of the target data is wide and is an effect of the data trend. Multi-imputation is a method of selecting the most suitable rule among multiple rules through simulation. The approach has a higher accuracy than single imputation because of testing multiple imputation methods using various sampling data. However, the process is time-consuming.

In automated processing, it is difficult to generate a high-accuracy model when the data pattern cannot be defined, while the imputed data can be calculated automatically.

### 2.3 PBE

The PBE [2], [9] approach is an automated transformation method with more than one example for input and output. The model infers a rule from inputted examples and generates other values in the same data using the same rule. PBE has three main

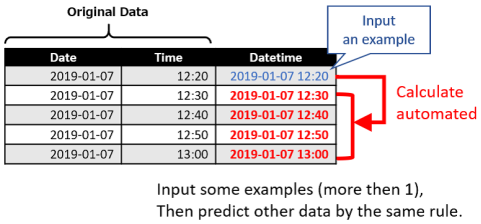


**Fig. 3** Example for PBE: generate *Datetime* from *Date* and *Time*.

**Table 1** Comparison of manual, automated, and proposed methods.

| | Manual processing | Automated processing | Proposed approach |
|---|---|---|---|
| Customization | Easy | Difficult | **Easy** |
| Automated Work | Nothing | All | **Almost** |
| Accuracy | Normal (*) | High | **High** |

(*) however, what we can work manually is limited.

processes: 1) a search algorithm that can efficiently search a match rule from the input data provided by analysts, 2) a ranking program to choose the most suitable process that satisfies the input examples provided by the analysts, and 3) an interaction model to facilitate usability for analysts [2]. This paper proposes the PBE approach for selecting the imputation method. Although it is too difficult to write the desired macros or scripts by the analysts who are not familiar with programming [9], PBE can help reduce the number of tasks of analysts without knowledge. An example of PBE is shown in **Fig. 3**. Now, we have *Date* column and *Time* column. If "*Date* + *Time*" format data is required, we input "2019-01-07 12:20" into the *Datetime* column in the first line. Then, the model infers that rule and automatically inputs the data into other rows using the inferred rule "*Date* + *Time*".

Recently, PBE has been focused on as a data transformation method for big data. The PBE approach uses data extraction, transformation, or formatting, such as Flash Fill [10] and FlashNormalize [11], and code transformation, such as Foofah [6]. PBE can integrate machine learning, and is more efficient for general code refactoring, application migration, and noise detection [2], [12], [13].

Our target is to reduce the number of analyst tasks while maintaining high-accuracy results. Therefore, we combine PBE and machine learning via Bayesian inference, inputting the human knowledge to the proposed processing as examples and recommending the most appropriate method of handling outliers and missing data by machine learning via Bayesian inference. If learning from one or a few examples, we might miss the correct result [14]. Correctly handling ambiguous problems is the strength of Bayesian inference.

Using the PBE approach, we propose an integration "Customization (of the model)", which is the advantage of manual processing to "automated work (reduce the analysts' workload)", and "accuracy", which is the main advantage of automated processing. Manual processing, automated processing, and the proposed processing (APREP-S) are summarized in **Table 1**.

## 3. Workflow for APREP-S

### 3.1 Overview of APREP-S Workflow

We propose a model termed APREP-S, which initially defines the imputation methods, and subsequently, calculates the proportion of the likelihood of each method and makes recommendations to the analyst. The APREP-S workflow comprises the analysts' tasks and automated tasks. The entire workflow of APREP-S is shown in **Fig. 4**. The analysis of APREP-S needs three phases: the analysis preparation phase, the model training phase, and the operation phase. A detailed flow of APREP-S is described in Section 4. Here, we show the analysis tasks involved in this workflow.

First, in the analysis preparation phase, the analyst calculates the rate of similarity of each dataset to decide the training data for target imputation data, which indicates the position of outliers and missing data. We use "k-Shape", which is a clustering method based on the similarity of time-series data. The analyst inputs multiple time-series data, including the data that the analyst wants to impute, to k-Shape. Then, k-Shape returns the classified cluster number of each data. The analyst selects one data in the same cluster with the target imputation data as training data. We describe k-Shape in Section 3.2.

Next, in the model training phase, the analyst inputs the training data selected in the analysis preparation phase to APREP-S. Then, APREP-S generates the model. After this phase, the ana-

**Algorithm 1** k-Shape algorithm: adapted from [15] Algorithm 3.

**INPUT:** $X$ is an $n$-by-$m$ matrix containing $n$ time-series of length $m$ are initially $z$-normalized.
$k$ is the number of clusters to produce.
**OUTPUT:** $IDX$ is an $n$-by-1 vector containing the assignment of $n$ time-series to $k$ clusters (initialized randomly).
$C$ is a $k$-by-$m$ matrix containing $k$ centroids of length $m$ (initialized as vectors with all zeros)

```
iter ← 0
IDX′ ← []
while IDX! = IDX′ and iter < 100 do
  IDX′ ← IDX
  // Refinement step
  for j ← 1 to k do
    X′ ← []
    for i ← 1 to n do
      if IDX(i) = j then
        X′ ← [X′; X(i)]
      end if
    end for
    C(j) ← ShapeExtraction(X′, C(j))(*)
  end for
  // Assignment step
  for i ← 1 to n do
    mindist ← ∞
    for j ← 1 to k do
      [dist, x′] ← SBD(C(j), X(i))
      if dist < mindist then
        mindist ← dist
        IDX(i) ← j
      end if
    end for
  end for
  iter ← iter + 1
end while
```

(*) "*ShapeExtraction*()" that is reffer to Algorithm 2 in paper [15].

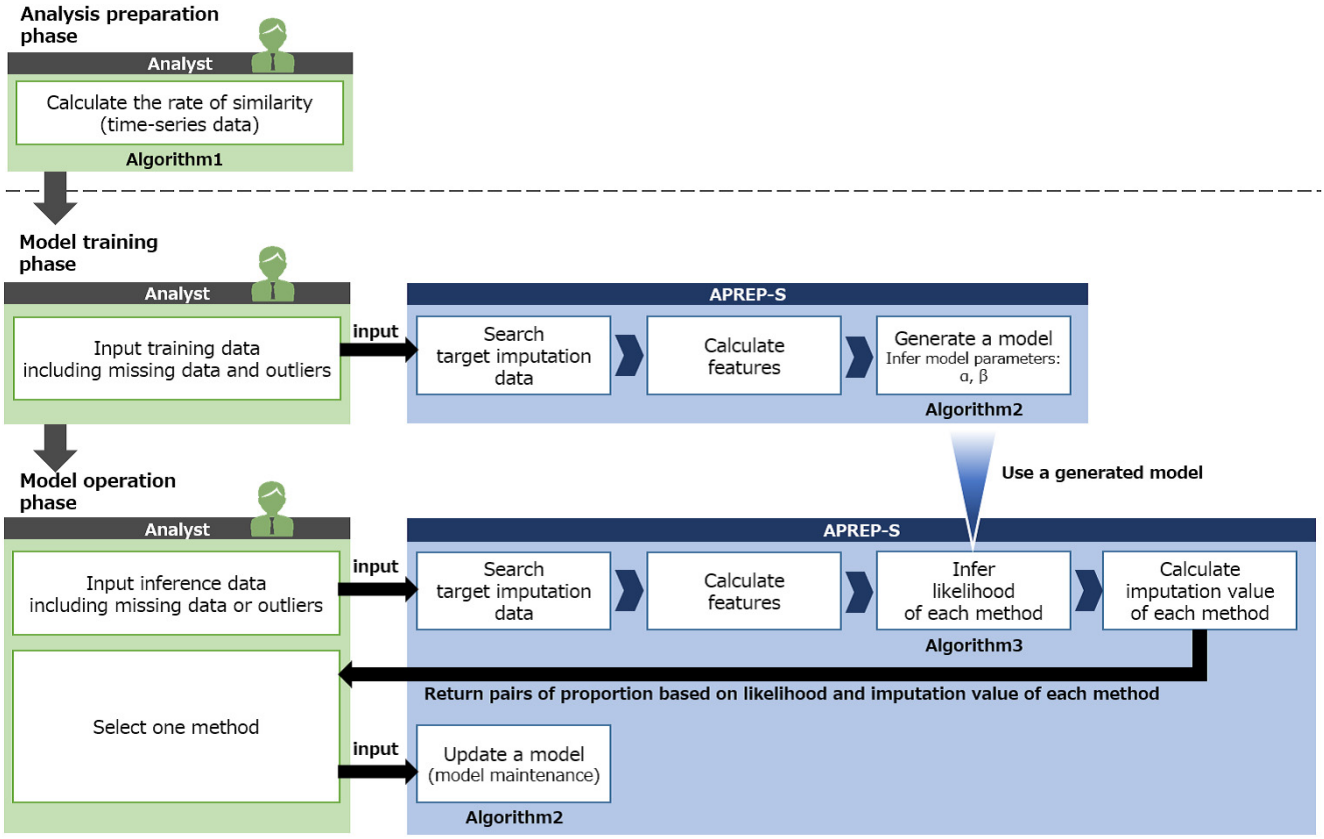


**Fig. 4** Workflow of the analyst and APREP-S: The green area denotes the analyst's tasks and the blue area denotes the APREP-S tasks. Our proposal is the model training phase and the model operation phase. The analyst uses an existing method "k-Shape" in the analysis preparation phase.

lyst can use the APREP-S model.

Last, in the model operation phase, the analyst operates data imputing by using the APREP-S model generated in the model training phase. This phase includes the model maintenance flow. The analyst inputs the inference data, which has outliers or missing data, to APREP-S. Then, APREP-S returns the pairs of recommendation proportion and the imputation value that refers to input data instead of outliers and missing data. The proportions and values are calculated using the same imputation method, and the multiple methods are defined in APREP-S. The analyst can select one imputation method, checking whether the proportions and values are appropriate. After that, the analyst inputs the selected method to APREP-S, and then, APREP-S updates a model based on the data added by the selected method. When the analyst inputs other target imputation data, APREP-S uses an updated model. For iterating this operation phase, the model accuracy improves.

### 3.2 k-Shape

We use the "k-Shape [15]" method for evaluating the data similarity. This method considers the shape of the time series in clustering tasks, in contrast to traditional methods such as k-means. It treats the observations in time-series data as independent attributes. In general, we consider the invariance of data before clustering, e.g., amplitude scaling, time-shifting, data length scaling, or occlusion. The k-Shape method is focused on amplitude scaling invariance and time-shifting invariance. Moreover, it does not depend on the domain form because it calculates cross-correlation by using normalized data as the distance measure for the similarity of data in clustering. It uses an independent method, named "shape-based distance (SBD)", by considering scaling and shifting with normalized data. SBD is expressed as

$$SBD(\boldsymbol{x},\boldsymbol{y}) = 1 - \max_{\omega}\left(\frac{CC_{\omega}(\boldsymbol{x},\boldsymbol{y})}{\sqrt{R_0(\boldsymbol{x},\boldsymbol{x})\cdot R_0(\boldsymbol{y},\boldsymbol{y})}}\right) \tag{1}$$

Let $\boldsymbol{x} = (x_1, ..., x_m)$ and $\boldsymbol{y} = (y_1, ..., y_m)$ be sequences, and

$$CC_{\omega}(\boldsymbol{x},\boldsymbol{y}) = R_{\omega-m}(\boldsymbol{x},\boldsymbol{y}), \quad \omega \in 1, 2, ..., 2m-1 \tag{2}$$

$$R_k(\boldsymbol{x},\boldsymbol{y}) = \begin{cases} \sum_{l=1}^{m-k} x_{l+k}\cdot y_l, & k \geq 0 \\ R_{-k}(\boldsymbol{y},\boldsymbol{x}), & k < 0 \end{cases} \tag{3}$$

k-Shape's algorithm refers to Algorithm 1. This algorithm is adapted from the paper [15] Algorithm 3. Function "ShapeExtraction(X, C)" refers to paper [15] Algorithm 2. This algorithm extracts the shape of time-series data.

The k-Shape method computes the centroid of the cluster and compares it with each time-series data. The data are classified into the closest cluster. The centroid is recalculated whenever a new time-series data joins. The k-Shape method iterates the calculation of the centroid until the cluster membership does not change.

## 4. Proposed Model - APREP-S

We propose a model termed APREP-S, whose workflow is

**Algorithm 2** Generation of APREP-S model (model training phase)

**INPUT:** $tr\_m$ is a selected method number list.
$\boldsymbol{x}$ is a list of normalized features which are calculated from TRN_LIST.
$Q$ is the number of features.
**OUTPUT:** APREP-S model

$\alpha \leftarrow$ Gaussian distribution $(\mu_\alpha, \sigma_\alpha)$
$\boldsymbol{\beta}$ ($Q$-by-$K$ matrix containing) $\leftarrow$ Gaussian distribution $(\mu_\beta, \sigma_\beta)$
**for** $k \leftarrow 1$ *to* $K$ **do**
  **for** $q \leftarrow 1$ *to* $Q$ **do**
    $\boldsymbol{y} = \alpha + \boldsymbol{\beta}\, x_q \leftarrow \alpha, \boldsymbol{\beta}$
    $p(m_k|\boldsymbol{y}) = \exp(\boldsymbol{y}(x_q))/\sum_{i=1}^{K} \exp(\boldsymbol{y}(x_i)) \leftarrow m_k, \boldsymbol{y}$
    $p(\boldsymbol{y}|m_k) \leftarrow tr\_m$
    $C(m_k|\boldsymbol{y}) \leftarrow \boldsymbol{y}$, $p(\boldsymbol{y}|m_k)$
    $\alpha_p, \boldsymbol{\beta}_p \leftarrow$ sampling with $C(m_k|\boldsymbol{y})$
  **end for**
**end for**
$\boldsymbol{y} \leftarrow \alpha_p, \boldsymbol{\beta}_p$
APREP-S model $\leftarrow \boldsymbol{y}$

shown in Fig. 4. In this section, we show the APREP-S tasks. In the model training phase, after receiving training data from the analyst, APREP-S searches the target imputation data on the training data. Then, APREP-S calculates the features and normalizes them. Last, APREP-S generates a model that has two parameters: $\alpha$ and $\beta$. The specific tasks and algorithm (Algorithm 2) of this generated model are shown in Section 4.3.1. In the model operation phase, after receiving target imputation data, APREP-S searches for target imputation data on received data. Then, APREP-S calculates features and normalizes them. Next, it infers the likelihood of each method using the generated model in the model training phase. APREP-S has some imputation methods of outliers or missing data. Therefore, APREP-S calculates the recommendation proportions of each method from the features and two model parameters $\alpha$ and $\beta$. Then, APREP-S calculates the imputation value of each method. Last, APREP-S returns the pairs of recommendation proportions (means likelihood) and imputation values of each method to the analyst. Moreover, the analyst inputs the selected method to APREP-S, which then updates the model. When the analyst inputs other data next, APREP-S uses the updated model. The accuracy of the model improves through the iteration of these flows. The specific tasks and algorithm (Algorithm 3) of this generated model are shown in Section 4.3.2.

**Figure 5** is a diagrammatic representation of APREP-S. It shows an example of a case in which the analyst inputs inference time-series data, indicated by a red line. The red line data have four outliers or missing data, that is, four target imputations, on this target imputation data. In addition, the black dashed line data in the training data of the red line, both of which are similar data. The black dashed line has three outliers or missing data, all of which have already decided the imputation method: the first one is method2, the second one is method1, and the third one is method1. As APREP-S has three imputation methods in this example, it returns three pairs of recommendation proportion and imputation value based on the training data. The analyst can select one suitable method, checking the recommendation proportions and imputation values.

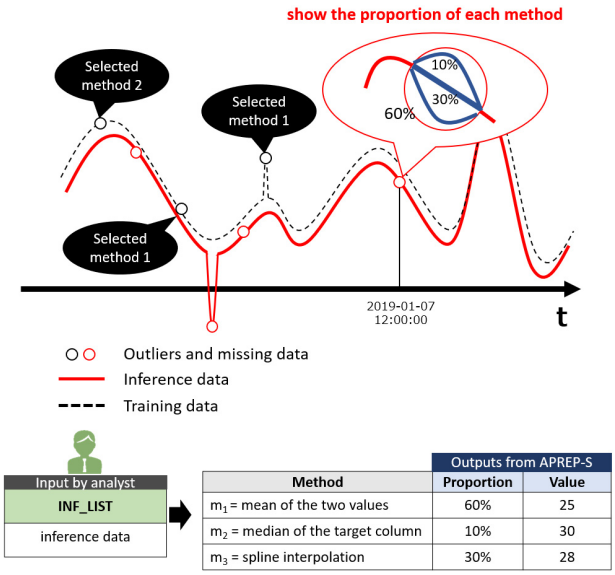


**Fig. 5** Example of APREP-S: the red line denotes target imputation data, the black dashed line indicates training data, and the black or red circle indicates outliers or missing data. The red line has four red circles and each black circle on the training data has already decided an imputation method. APREP-S has three imputation methods.

The features of APREP-S are described as follows.

### 4.1 Formalism of our Approach

In general, a time-series data provides pairs of values: $\boldsymbol{O} = (t_1, o_1), ..., (t_N, o_N)$, $\boldsymbol{t}$ means time and $\boldsymbol{o}$ means observation values. Assuming sensor data, some data $o_i$ $(1 \leq i \leq N)$ are outliers or missing data. Our goal is that APREP-S outputs the pair of the recommendation proportion $\boldsymbol{P}$ and imputation value $\boldsymbol{v}$ of each method $m_1, m_2 ... \in \boldsymbol{M}$ ($m$ is finite) when the analyst inputs $\boldsymbol{O}$. $TR\_LIST, INF\_LIST \in \boldsymbol{O}$: $TR\_LIST$ indicates the data that the analyst inputs in the model training phase, and $tr$ is a list of target imputation data extracted from $TR\_LIST$. $INF\_LIST$ indicates data that the analyst inputs in the model operation phase, and $inf$ is a list of target imputation data extracted from $INF\_LIST$. APREP-S needs a method number list of $tr$. $tr\_m$ is a list that stores the method number that an analyst selects for each element of $tr$.

In case of an example Fig. 5, $TR\_LIST$ is a black dash line and $tr\_m$=(2, 1, 1), $INF\_LIST$ is a red line. The method $M$ has $m_1$="Mean of the two values", $m_2$="Median of the target column", $m_3$="Spline interpolation". The outputs of APREP-S are $\boldsymbol{P}$=(60%, 10%, 30%), $\boldsymbol{v}$=(25, 30, 28).

### 4.2 Probability Model

In this paper, we infer the probability of each imputation method $m_k \in \boldsymbol{M}$ that APREP-S has. APREP-S calculates each likelihood based on Bayesian inference. As $\boldsymbol{M}$ is a discrete value, $\boldsymbol{p}$ can be calculated from the proportion of likelihoods in $\boldsymbol{M}$ in each $m_i$. The posterior probability of $p(m_k|\boldsymbol{y})$ is based on Bayesian inference [16], [17]:

$$p(m_k|\boldsymbol{y}) = \frac{p(\boldsymbol{y}|m_k)p(m_k)}{\sum_{i=1}^{K} p(\boldsymbol{y}|m_i)p(m_i)} = \frac{\exp(\boldsymbol{y}(x_k))}{\sum_{i=1}^{K}\exp(\boldsymbol{y}(x_i))} \quad (4)$$

$$\boldsymbol{y}(x_q) = \alpha + \boldsymbol{\beta}\, x_q \quad (1 \le q \le Q) \quad (5)$$

Let $x_1, x_2, ... \in \boldsymbol{x}$($x$ is finite) be a set of features for the APREP-S model, and $m_1, m_2, ... \in \boldsymbol{M}$ ($m$ is finite) be a method type defined in APREP-S. $p(\boldsymbol{y}|m_k)$ is a prior distribution of each method, which can be calculated from $tr\_m$ at first in the model training phase. $\alpha$ and $\boldsymbol{\beta}$ are the parameters of the APREP-S model. In Eq. (4), we use the Softmax function as $\boldsymbol{y}$ because the number of methods will be naturally more than two. Each method has a $\boldsymbol{y}$, which is also a parameter of the likelihood function of APREP-S. The likelihood function is

$$C(\boldsymbol{M}|\boldsymbol{y}) = \prod_{k=1}^{K}(y_k^{u_k}) \quad (6)$$

Let $u_k$ denote the probability that the method is $m_k$, $0 \le y_k \le 1$, and $\sum_k y_k = 1$. Given $p(m_k|\boldsymbol{y})$ is a normalized exponential function, because $\sum_{i=1}^{K} p(m_i|\boldsymbol{y}) = \sum_{i=1}^{K} u_i = 1$. Hence, the likelihood (Eq. (4)) equals a proportion $\boldsymbol{P}$.

### 4.3 APREP-S Model

#### 4.3.1 Model Training Phase: Parameters $\alpha, \boldsymbol{\beta}$

In the model training phase, APREP-S has the main task, termed "generate a model". During training, the model learns the parameters $\alpha$ and $\boldsymbol{\beta}$, which characterize the conditional probability of a program given the input $p(m_k|\boldsymbol{y})$. The algorithm refers to Algorithm 2 with the following specific flow:

( 1 ) Input two data: a list $tr\_m$ of each imputation method number extracted from $TR\_LIST$ in "search target imputation data" and a normalized feature list $\boldsymbol{x}$ generated in "calculate features".
( 2 ) Set two probability parameters $\alpha$ and $\boldsymbol{\beta}$, where $\boldsymbol{\beta}$ is a $Q$-by-$K$ matrix, $Q$ is the number of elements of $\boldsymbol{x}$, $K$ is the number of types of methods $\boldsymbol{M} \ni m_1, m_2, ..., m_K$. Both parameters $\alpha$ and $\boldsymbol{\beta}$ are Gaussian distribution.
( 3 ) Define a linear function of $x$: $\boldsymbol{y}$ (Eq. (5)) with $\alpha$ and $\boldsymbol{\beta}$.
( 4 ) For each $m \in \boldsymbol{M}$, define a posterior probability: $p(m|\boldsymbol{y})$ (Eq. (4))
( 5 ) For each $m \in \boldsymbol{M}$, define a prior probability: $p(\boldsymbol{y}|m)$ from a likelihood function $C(\boldsymbol{M}|\boldsymbol{y})$ (Eq. (6)). The first prior probability is calculated from a method number list $tr\_m$.
( 6 ) Sampling $\boldsymbol{y}$ including $\alpha$, and $\boldsymbol{\beta}$ with the $C(\boldsymbol{M}|\boldsymbol{y})$.
( 7 ) Define $\boldsymbol{y}$ having posterior distributions $\alpha_p$, and $\boldsymbol{\beta}_p$ as the APREP-S model.

Therefore, the APREP model for each method $m_i$ is

$$p(m_k|\boldsymbol{y}) = \frac{\exp(\alpha_p + \boldsymbol{\beta}_p m_k)}{\sum_{q=1}^{Q}\exp(\alpha_p + \boldsymbol{\beta}_p m_q)} \quad (7)$$

#### 4.3.2 Model Operation Phase: Inference with APREP-S model

In the model operation phase, APREP-S has the main task termed "infer likelihood of each method". In this paper, we define $\alpha$ and $\boldsymbol{\beta}$ as fixed values of the mean of each method.

**Algorithm 3** Calculation of the recommendation proportion of each method based on likelihood (model operation phase)

**INPUT:** $\boldsymbol{x}$ is a list of normalized features calculated from $INF\_LIST$.
$Q$ is the number of features.
APREP-S model is a generated model in Algorithm 2.
$\alpha_p, \boldsymbol{\beta}_p$ are parameters of APREP-S model.
**OUTPUT:** proportion $\boldsymbol{P}$ (unit:%)

```
y ← APREP-S
for k ← 1 to K do
    for q ← 1 to Q do
        E_α[α|m_k] = Σ_q p(α_p|m_k)α_p
        α ← E_α[α|m_k]
        E_β[β|x_q, m_k] = Σ_q p(β_p|m_k, x_q)β_p
        β ← E_β[β|x_q, m_k]
    end for
    y = α + βx ← α, β
    p(m_k|y) = exp(α + βx)/ Σ_{k=1}^{K} exp(α + βx) ← x
    P_k(m_k|y) = 100 * p(m_k|y)
end for
```

First, APREP-S calculates the expectations of each parameter, and then, calculates the likelihoods of each method for obtaining the recommendation proportion. The algorithm refers to Algorithm 3 and the following specific flow:

( 1 ) Input two data: a normalized feature list $\boldsymbol{x}$ generated in "calculate features" and an APREP-S model generated in the model training phase.
( 2 ) Calculate each expectation as fixed values $\mathbb{E}_\alpha$ and $\mathbb{E}_\beta$ using $\alpha_p$ and $\boldsymbol{\beta}_p$.
( 3 ) Define the APREP-S model for inference $\boldsymbol{y}$ with $\mathbb{E}_\alpha$ and $\mathbb{E}_\beta$
( 4 ) For each element of $inf$ and each $m \in \boldsymbol{M}$, calculate likelihoods $p(m|\boldsymbol{y})$ using the APREP-S model.
( 5 ) Calculate the recommendation proportions based on the likelihoods of each method.

## 5. Evaluation

We evaluate the "model training phase" and "model operation phase" in Fig. 4.

In this evaluation, we compare the sum of the square of the errors ($ERR$) between $\boldsymbol{Org}$ and APREP-S output values $\boldsymbol{v}$, and between $\boldsymbol{Org}$ and the values calculated using other methods: i) mean of the entire data, ii) mean of the around-the-target imputation data, and iii) cubic spline interpolation. The specific is shown in Section 5.2.3. The model that corresponds to a smaller $ERR$ is the one with higher accuracy. We create a dataset with outliers and missing data based on the original data and let the original data [18] list be $\boldsymbol{Org}$=($org_1, org_2, ..., org_N$). If the number of target imputation data is $o_i \in \boldsymbol{O}$ (i=1,...,N, $o_i$ is a data that the analyst inputs), the APREP-S model returns imputation values $v_i$. Therefore, ERR is given as

$$ERR = \frac{1}{2}\sum_{i=1}^{I}(org_i - v_i)^2 \quad (8)$$

### 5.1 Evaluation Preparation

Before the evaluation, we need to define a dataset (including calculating data similarity), methods, and features. Each specific

is described as follows.

#### 5.1.1 Evaluation Dataset

We use a dataset [18] composed of wireless temperature and humidity sensors (DHT-22) installed inside or outside a home. The well-known sensors measure pressure, temperature, humidity, magnetometer, gyroscope, accelerometer, image, etc. In this evaluation, we select temperature and humidity as popular sensor data because they are numerical and time-series data having daily trends.

This dataset has 29 columns, e.g., measurement time, temperature, humidity, pressure, and wind speed. The temperature and humidity columns have nine sensors each, installed on the first floor, second floor, and outside, e.g., sensor1 measures temperature T1 and humidity RH1. There are four sensors on the first floor; sensor1 is in the kitchen area, sensor2 is in the living area, sensor3 is in the laundry room, and sensor4 is in the office room. Sensor1 and sensor2 are in the same room. There are five sensors on the second floor; sensor5 is in the bathroom, sensor6 is at the north side outside the house, sensor7 is in the ironing room, sensor8 is in the children's room, and sensor9 is in the parents' room. The time span of the original dataset is 137 days (4.5 months), with 19,735 rows per sensor. Each sensor transmits data approximately every 3.3 min, which are then aggregated from 3.3 to 10 min. The digital DHT-22 sensors used have an accuracy of ± 0.5°C for temperature measurements and ± 3% for relative humidity. We create evaluation data including outliers and missing data based on this dataset. Let the occurrence probability of missing data depend on the exponential distribution

$$f(e) = \frac{1}{\epsilon}\exp\left(-\frac{e}{\epsilon}\right) \quad (500 \le \epsilon \le 1000) \tag{9}$$

and define nine outliers and one missing data out of every 10 datasets. The outlier difference between the original data and the evaluation data depends on a Gaussian distribution.

$$f(e) = \frac{1}{\sqrt{2\pi\sigma^2}}\exp\left\{-\frac{(e-org_i)^2}{2\sigma^2}\right\} \quad (0 \le \sigma^2 \le 10) \tag{10}$$

Let $\sigma^2$ be the variance.

We calculate the rate of similarity of evaluation data using "k-Shape" for classification, as mentioned in Section 3. This is an analysis preparation phase in Fig. 4. In this paper, we extract every 30 min of data from the original dataset for calculating similarity. The results are shown in **Fig. 6** and **Fig. 7**. First, we need to decide a cluster number to be inputted to k-Shape. Therefore, we use an elbow chart for deciding the number of clusters in a chart (a). As elbow charts, the best $T$'s cluster number is three and $RH$'s cluster number is four. The k-Shape results are indicated by the line graph (b). $T$ is classified as $cluster1 = [T2, T6]$, $cluster2 = [T1, T3]$, $cluster3 = [T4, T5, T7, T8, T9]$. $RH$ is classified as $cluster1 = [RH3, RH4, RH7, RH8, RH9]$, $cluster2 = [RH6]$, $cluster3 = [RH1, RH2]$, $cluster4 = [RH5]$. In addition, we calculate DTW [19], [20], which detects patterns in a data stream or time series by the distance measure between the data. The distance of DTW is shown as a heat map (c). A deep blue color indicates a large difference, while a light color indicates a small difference between the two data. However, $T$'s heat map is without $T6$, while $RH$'s heat map is without $RH5$ and $RH6$, because the differences are too large between them and the other data because of $T6$, and $RH6$ is installed outside and $RH5$ is installed in the bathroom.

We choose three pairs of data: $[tr, inf]$=[$T1$, $T3$], [$RH1$, $RH2$], [$RH2$, $RH1$], and [$RH3$, $RH4$]. Each pair is classified in the same cluster by the k-Shape, and indicated by a light color in the DTW heat map. $T1$ is configured to have 20 outliers and missing data with Eq. (9) and Eq. (10), $T3$ has 39, $RH1$ has 36, $RH2$ has 37, $RH3$ has 20, and $RH4$ has 38.

#### 5.1.2 Evaluation Methods

We define three methods $m_1, m_2, m_3 \in \boldsymbol{M}$ for evaluation in APREP-S. $m_1$ is a mean of front and behind, $m_2$ is an imputation of the front data without the transform, and $m_3$ is a cubic spline interpolation [21]. In this evaluation, we define single imputation methods because it is a well-used imputation method.

- $m_1(o_i) = (o_{i-1} + o_{i+1})/2$
- $m_2(o_i) = o_{i-1}$
- $m_3(o_i) = a_j(o_i - o_j)^3 + b_j(o_i - o_j)^2 + c_j(o_i - o_j) + d_j$ $(1 \le j \le N-1)$

Let $i$ $(i = 1, ..., N)$ be a target imputation data in $tr$ or $inf$.

#### 5.1.3 Evaluation Features

We define three features $x_1, x_2, x_3 \in \boldsymbol{x}$. $x_1$ is the gradient of the front and before data, $x_2$ is the trend of the two front data, and $x_3$ is the difference between the mean of the front and the behind and that of all data. Let $i$ $(0 \le i \le N)$ be one of the target imputation data in $tr$ or $inf$.

- $x_1(o_i) = (o_{i-1} + o_{i+1})$
- $x_2(o_i) = (o_{i-2} + o_{i-1})$
- $x_3(o_i) = |(o_{i-1} + o_{i+1})/2| - \text{mean}(\boldsymbol{o})$

### 5.2 Evaluation Measuring

#### 5.2.1 Model Training Phase

The two parameters of APREP-S $\alpha$ and $\boldsymbol{\beta}$ depend on Gaussian distribution (mean: $\mu$=0, variance: $\sigma^2$=2). In addition, an analyst inputs a selected method number list $tr\_m$, which is shown below for each imputation method on the target imputation data. Each $tr$ has each $tr\_m$, e.g., $T1$ has (3, 1, 1, 3, 2, 3, 1, 1, 2, 2, 3, 2, 3, 2, 3, 3, 3, 1, 3, 3) (the list size is 20). The flow of generating the APREP-S model is as follows:

( 1 ) Searching outliers and missing data (target imputation data) and creating method number list $tr\_m$
( 2 ) Calculating features of target imputations: $x_1, x_2, x_3$, and normalizing $x_1, x_2, x_3$
( 3 ) Infer model parameters $\alpha$ and $\boldsymbol{\beta}$ by using Algorithm 2. Input data are $tr\_m$ list, methods $\boldsymbol{M}$, features $\boldsymbol{x}$. Output is APREP-S model.

#### 5.2.2 Model Operation Phase

For data $inf$, we search target imputations and calculate features as well as $tr$. Then, we infer the likelihood of each method for each $inf$ by using each APREP-S model. For example, we infer the imputation method of $T3$ using the APREP-S model, which generates $T1$, as $inf$=$T3$ is a pair of $tr$=$T1$. The recommendation proportions $\boldsymbol{P}$ of the first three imputation targets are as follows.

1st: $m_1 = 39.67\%$, $m_2 = 4.97\%$, $m_3 = 55.36\%$

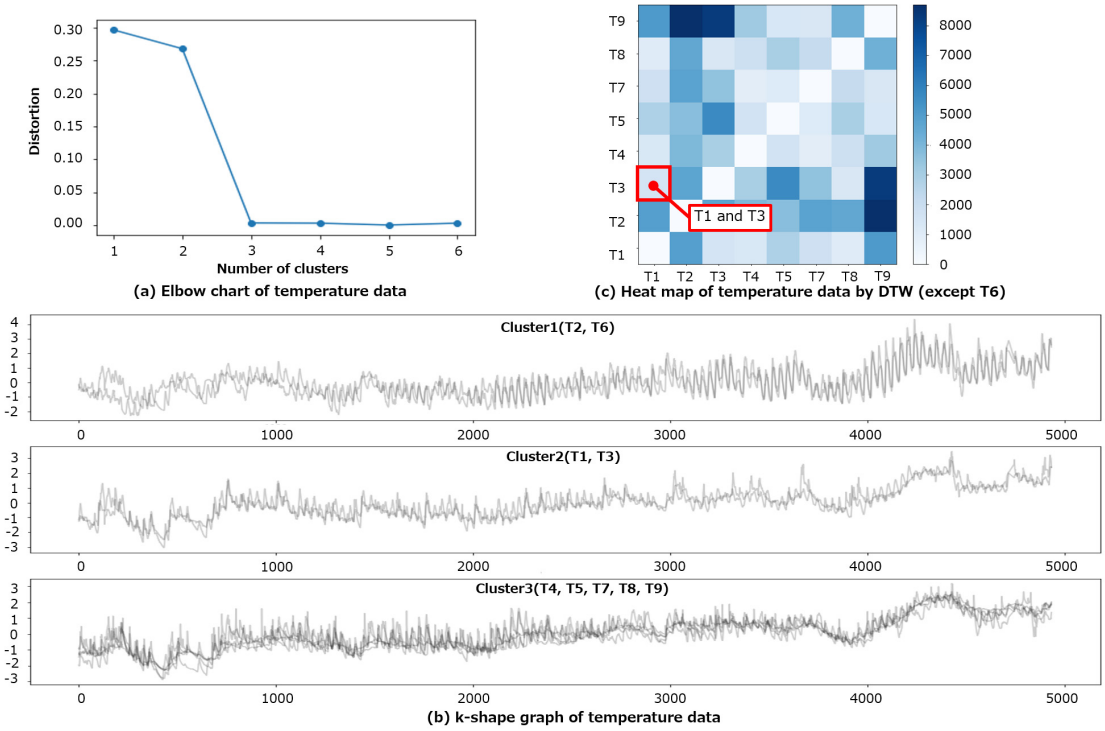


**Fig. 6** Result of similarity of temperature ($T$) data: (a) elbow chart, (b) classification of k-Shape: $T$ data classified three clusters, (c) heat map of dynamic time warping (DTW) (deep blue color denotes a large difference, while light blue color denotes a small difference).

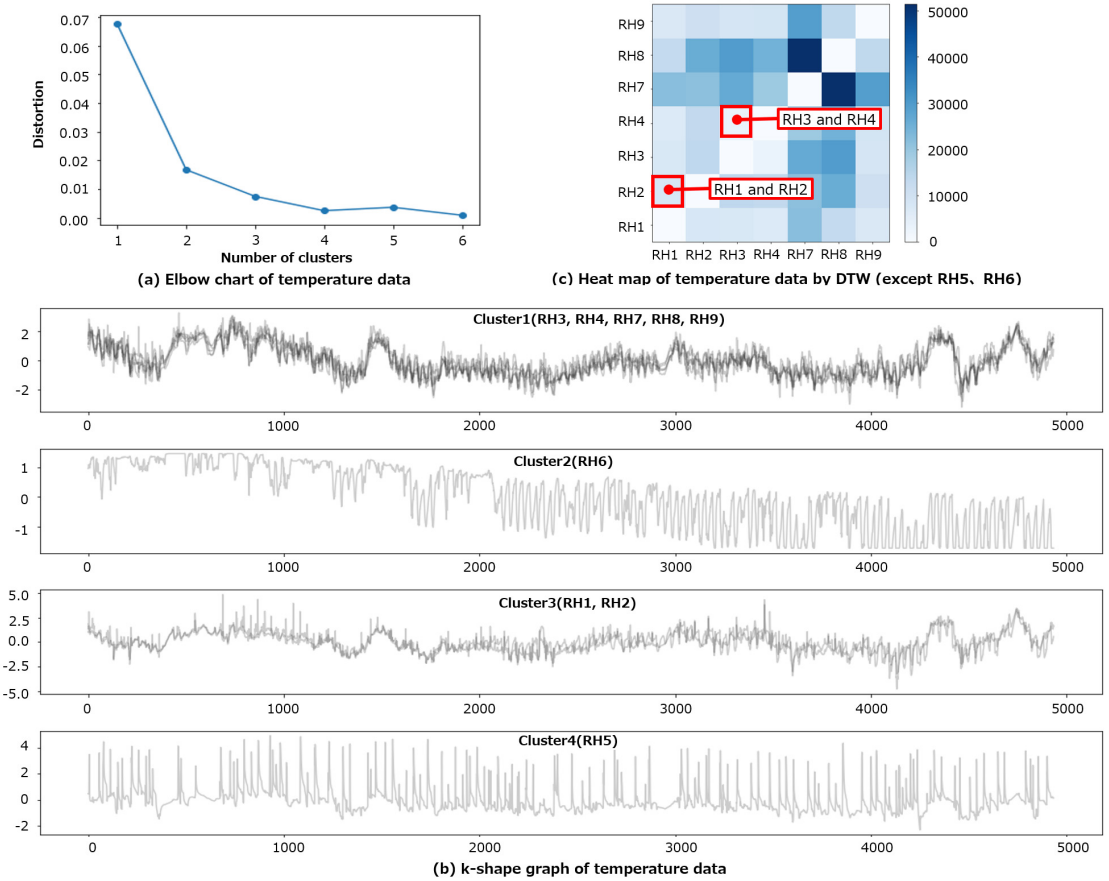


**Fig. 7** Result of similarity of humidity ($RH$) data: (a) elbow chart, (b) classification of k-Shape: $RH$ data are classified into four clusters, (c) heat map of DTW (deep blue color denotes a large difference, while light blue color denotes a small difference).

2nd: $m_1 = 9.02\%, m_2 = 0.22\%, m_3 = 90.76\%$
3rd: $m_1 = 12.23\%, m_2 = 0.22\%, m_3 = 87.55\%$

Then, APREP-S calculates imputation values of each method. The values $\boldsymbol{v}$ of the first three imputation targets are as follows.
1st: $m_1 = 20.60, m_2 = 20.60, m_3 = 21.00$
2nd: $m_1 = 19.34, m_2 = 19.29, m_3 = 19.70$
3rd: $m_1 = 20.13, m_2 = 20.20, m_3 = 19.78$

In this evaluation, we assume that an analyst selects a method of the highest probability. As a result of inference $T3$ by APREP-S, the selected method number list of $T3$ is $inf\_m$=(3, 3, 3, 1, 1, 2, 3, ..., 3, 3) (the list size is 39).

#### 5.2.3 Comparing Model

We define three single imputation methods for comparing with APREP-S: (i) Mean of the the entire data = mean($\boldsymbol{o}$), (ii) Mean of the around the target imputation data = mean($o_j$) ($i-36 \le j \le i+36$), (iii) Cubic spline interpolation=$a_j(o_i-o_j)^3+b_j(o_i-o_j)^2+c_j(o_i-o_j)+d_j$ ($i-36 \le j \le i+36$). We use a sensor data every 10 min (Section 5.1.1). Therefore, the range of the target data is defined for each of the 6 h before and after the target imputation data $o_i$ ($36 \times 2 + 1 = 73$ rows). "Mean of the entire data" inputs the mean of all $inf$ data. "Mean of the around-the-target imputation data" inputs the mean of the 6 h before and after the target imputation data. This corresponds to 12 h, 72 rows. "Cubic spline interpolation" inputs the median of the list that has 73 rows from the model that learns based on the original 72 rows of data.

### 5.3 Evaluation Result

The result of the accuracy is shown in **Table 2**. We calculate the sum of squares error (Eq. (8)) for each method - APREP-S, mean of the entire data, mean of the around the target imputation data, spline interpolate, and original data. In the inference of $RH2$ based on $RH1$, $RH1$ based on $RH2$, and $RH4$ based on $RH3$, the highest-accuracy method is APREP-S. In the inference of $T1$ and $T3$ pair, the highest-accuracy method is Spline, and that with the third highest accuracy is APREP-S. Single imputation has the worst accuracy in each $tr$ and $inf$ pairs. Therefore, APREP-S is the most suitable method for $RH$ data, but not the best method for $T$ data. We discuss this reason in Section 5.4.

### 5.4 Evaluation Discussion

We consider that APREP-S is suited for data which has more changing points. The number of changing data and the percentage are shown in **Table 3**. There are more target imputation data at the changing point in $RH$ data than $T$ data; for example, $T3$ has only two changing points in the target imputation data, whereas $RH1$ has seven changing points. In this evaluation, we create the imputation value at random, as mentioned in Section 5.1.1. As $RH$ data is more fluctuating than $T$ data, the possibility that the changing points become the target imputation data is higher than that of becoming the $T$ data. The line graph of $RH$ and $T$ data during one week is shown in **Fig. 8**. The above lines indicate $RH$ data, while the below lines indicate $T$ data. As a result, although the accuracies of spline interpolation and mean are enough for gentle data, they are not suitable methods as the imputation of the changing point. On the other hand, the accuracy of APREP-S is not too low for gentle data and the highest for imputation of the changing points. That is, we consider that APREP-S is suited for fluctuating data such as humidity data, human motion, and trajectory data.

**Table 2** Comparison of accuracy using sum of squares error (Eq. (8)).

| $tr$ | $inf$ | APREP-S | All | Around | Spline |
|---|---|---|---|---|---|
| $T1$ | $T3$ | 5.81 | 88.96 | 2.61 | 0.20 |
| $RH1$ | $RH2$ | 0.15 | 175.18 | 16.87 | 0.99 |
| $RH2$ | $RH1$ | 355.49 | 935.75 | 683.26 | 526.80 |
| $RH3$ | $RH4$ | 0.16 | 370.97 | 8.40 | 0.21 |

(*) "All" indicates Mean of the entire data
"Around" Mean of the around-the-target imputation data
"Spline" Cubic Spline Interpolation

**Table 3** Feature of evaluation data.

| | $RH1$ | $RH2$ | $RH3$ | $RH4$ | $T1$ | $T3$ |
|---|---|---|---|---|---|---|
| changing point(*) | 7/36 | 5/37 | 2/20 | 5/38 | 2/20 | 2/39 |
| percentage | 19% | 14% | 10% | 13% | 10% | 5% |

(*) changing point: *changing points / all imputation targets*

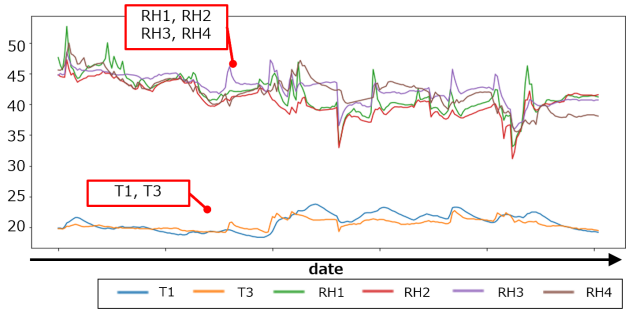


**Fig. 8** Line graph of $T$ and $RH$ data during a week.

## 6. Conclusion

This paper proposes APREP-S based on the PBE approach, which imputes values into data such as sensor data, including outliers and missing data. APREP-S integrates the advantages of manual processing, "customization", and those of automated processing, "automated work" and "accuracy".

The following are the conclusions of this paper:

- By comparing APREP-S with other imputation methods, it is verified that APREP-S based on the PBE approach is an effective imputation method.
- For generating the APREP-S model, we can use similarity data as training data.
- As an evaluation result, we consider that APREP-S tends to be more suitable for fluctuating data. It is accurate enough even for imputation by only mean and only spline interpolation in the gentle data.

We find that APREP-S is a suitable method for imputation of outliers and missing data. However, the effectiveness of APREP-S is slightly weak in gentle data, as indicated by the evaluation result. If we define a more suitable method for gentle data in APREP-S, the accuracy of APREP-S can be improved. Moreover, in the evaluation of this paper, we create imputation values at random, which are not continuous. We consider the application scope of APREP-S, not only non-continuous imputation but also area imputation, if we define the bulk imputation method as an imputation method that can be selected in APREP-S. Therefore we research these methods as a next step. In addition, "k-Shape", which is the method for checking the similarity of data in the

analysis preparation phase, needs to decide the number of clusters before classification. We assume that the analysis preparation phase can be included in the model training phase to use any method of classification without deciding the number of clusters.

## References


[1] Qi, Z., Wang, H., Li, J. and Gao, H.: Impacts of Dirty Data: and Experimental Evaluation, arXiv:1803.06071 [cs, stat] (2018).

[2] Gulwani, S. and Jain, P.: Programming by Examples: PL meets ML, *Microsoft Research* (2017) (online), available from ⟨https://www.microsoft.com/en-us/research/publication/programming-examples-pl-meets-ml/⟩.

[3] Nagashima, H. and Kato, Y.: APREP-DM: A Framework for Automating the Pre-Processing of a Sensor Data Analysis based on CRISP-DM, *PerFoT'19 - International Workshop on Pervasive Flow of Things* (*PerFoT'19*) (2019).

[4] Qi, J., Cui, T.G. and Martin, M.: OpenRefine, Supported by Google News Initiative (online), available from ⟨http://openrefine.org/⟩ (accessed 2019-05-05).

[5] Trifacta: Trifacta Wrangler, Trifacta (online), available from ⟨https://www.trifacta.com/start-wrangling/⟩ (accessed 2019-05-05).

[6] Jin, Z., Anderson, M.R., Cafarella, M. and Jagadish, H.V.: Foofah: Transforming Data By Example, *Proc. 2017 ACM International Conference on Management of Data, SIGMOD '17*, pp.683–698, ACM (online), DOI: 10.1145/3035918.3064034 (2017).

[7] Graham, J.W.: Missing Data Analysis: Making It Work in the Real World, *Annual Review of Psychology*, Vol.60, No.1, pp.549–576 (online), DOI: 10.1146/annurev.psych.58.110405.085530 (2009).

[8] Pedersen, A.B., Mikkelsen, E.M., Cronin-Fenton, D., Kristensen, N.R., Pham, T.M., Pedersen, L. and Petersen, I.: Missing data and multiple imputation in clinical epidemiological research, *Clinical Epidemiology*, Vol.9, pp.157–166 (2017) (online), available from ⟨https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5358992/⟩.

[9] Gulwani, S., Harris, W.R., Singh, R.: Spreadsheet Data Manipulation Using Examples, *Comm. ACM*, Vol.55, No.8, pp.97–105 (online), DOI: 10.1145/2240236.2240260 (2012).

[10] Gulwani, S.: Automating String Processing in Spreadsheets Using Input-output Examples, *Proc. 38th Annual ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages, POPL '11*, pp.317–330, ACM (online), DOI: 10.1145/1926385.1926423 (2011).

[11] Kini, D. and Gulwani, S.: FlashNormalize: Programming by Examples for Text Normalization, *IJCAI*, pp.776–783 (2015) (online), available from ⟨https://www.microsoft.com/en-us/research/publication/flashnormalize-programming-examples-text-normalization/⟩.

[12] Menon, A.K., Omer, T., Gulwani, S., Lampson, B. and Kalai, A.T.: A Machine Learning Framework for Programming by Example, *Proc. 30th International Conference on Machine Learning* (*ICML 2013*), Vol.28, No.1, pp.187–195 (2013) (online), available from ⟨https://www.microsoft.com/en-us/research/publication/machine-learning-framework-programming-example/⟩.

[13] Raychev, V., Bielik, P., Vechev, M. and Krause, A.: Learning Programs from Noisy Data, *Proc. 43rd Annual ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages, POPL '16*, pp.761–774, ACM (online), DOI: 10.1145/2837614.2837671 (2016).

[14] Ellis, K., Solar-Lezama, A. and Tenenbaum, J.: Sampling for Bayesian Program Learning, *Advances in Neural Information Processing Systems 29*, Lee, D.D., Sugiyama, M., Luxburg, U.V., Guyon, I. and Garnett, R. (Eds.), Curran Associates, Inc., pp.1297–1305 (2016) (online), available from ⟨http://papers.nips.cc/paper/6082-sampling-for-bayesian-program-learning.pdf⟩.

[15] Paparrizos, J. and Gravano, L.: k-Shape: Efficient and Accurate Clustering of Time Series, *Proc. 2015 ACM SIGMOD International Conference on Management of Data - SIGMOD '15*, pp.1855–1870, ACM Press (online), DOI: 10.1145/2723372.2737793 (2015).

[16] Bishop, C.M.: *Pattern recognition and machine learning*, Information science and statistics, Springer (2006).

[17] Davidson-Pilon, C.: *Bayesian Methods for Hackers: Probabilistic Programming and Bayesian Inference*, Addison-Wesley Professional, 1 edition (2015).

[18] Candanedo, L.M., Feldheim, V. and Deramaix, D.: Data driven prediction models of energy use of appliances in a low-energy house, *Energy and Buildings*, Vol.140, pp.81–97 (online), DOI: 10.1016/j.enbuild.2017.01.083 (2017).

[19] Berndt, D.J. and Clifford, J.: Using Dynamic Time Warping to Find Patterns in Time Series, *Proc. 3rd International Conference on Knowledge Discovery and Data Mining, AAAIWS'94*, Vol.3, pp.359–370, AAAI Press (1994) (online), available from ⟨http://dl.acm.org/citation.cfm?id=3000850.3000887⟩.

[20] Keogh, E.J. and Pazzani, M.J.: Derivative Dynamic Time Warping, *Proc. 2001 SIAM International Conference on Data Mining*, pp.1–11, Society for Industrial and Applied Mathematics (2001).

[21] Mckinley, S. and Levine, M.: Cubic Spline Interpolation, *Coll. Redw.*, Vol.45 (1999).


**Hiroko Nagashima** received her B.Sc. from Tokyo Woman's Christian University in 2009, and her Master of Technology degree from Advanced Institute of Industrial Technology in 2016. She is a Ph.D. student at Tokyo Woman's Christian University. Her current research focuses on data analysis techniques for IoT systems. She is a member of IPSJ, and IEEE.

**Yuka Kato** received her B.Sc. from the University of Tokyo in 1989 and her M.E. and Ph.D. from the University of Electro-Communications in 1999 and 2002. From 1989 to 1998, she was with NTT and engaged in research on traffic control in ATM networks. She was a research associate at the University of Electro-Communications from 2002 to 2006, an associate professor and a professor at Advanced Institute of Industrial Technology from 2006 to 2014. Since 2014, she is a professor at Tokyo Woman's Christian University. Her research interests include information networks, network robots, and mathematical models for robotics. She is a member of IPSJ, RSJ, ACM, and IEEE.