Data Imputation for Multivariate Time Series Sensor Data with Large Gaps of Missing Data

Rui Wu, Scott D. Hamshaw, Lei Yang, Dustin W. Kincaid, Randall Etheridge, Amir Ghasemkhani

***Abstract*— Imputation of missing sensor-collected data is often an important step prior to machine learning and statistical data analysis. One particular data imputation challenge is filling large data gaps when the only related data comes from the same sensor station. In this paper, we propose a framework to improve the popular multivariate imputation by chained equations (MICE) method for dealing with missing data. One key strategy we use to improve model accuracy is to reshape the original sensor data to leverage the correlation between the missing data and the observed data. We demonstrate our framework using data from continuous water quality monitoring stations in Vermont. Because of possible irregularly spaced peaks throughout the time series, the reshaped data is split into extreme and normal values and two MICE models are built. We also recommend that sensor-collected data should be transformed to meet the machine learning model assumptions. According to our experimental results, these strategies can improve MICE data imputation model accuracy at least 23% for large data gaps based on R^2 values and are promising to be applied for other data imputation algorithms.**

***Index Terms*— Data Imputation, Large Missing Data Gap, MICE, Multivariate, Time Series**

I. Introduction

Missing data is a common issue with sensor-collected datasets across domains including environmental monitoring, structural health monitoring, bioinformatics, and other Internet of Thing (IoT) applications. Data gaps can occur for various reasons, such as damaged sensors, loss of power, and problems with data storage or transmission. Data imputation is the process of replacing missing data with substituted values [1] and it is an important data pre-processing step for subsequent data-driven or physics-based modeling. In this paper, we analyze approaches to improve the accuracy of a data imputation model for multivariate time series sensor data with large data gaps and propose a data imputation framework.

Previous research has focused on **"how to predict the missing values within sensor collected data [2]–[5]."** One reasonable approach to gap-filling sensor data is to use data collected from a nearby sensor station and leverage the spatial autocorrelation found in many systems, especially environmental phenomena (e.g., weather, hydrology). For example, if multiple sensors are located within a study area it is possible to leverage data from a neighboring sensor(s) as reference(s) for data imputation [6], [7]. However, there are many scenarios where another dataset from nearby sensors is not available (e.g., when the budget limits the deployment of duplicate sensors or nearby sensors are damaged). In this scenario, the research question becomes **"How to predict the missing values using sensor data collected from the same sensor station"**, which is inherently more challenging. One approach that has been explored to address this challenge is the use of physics-based model output to impute missing sensor data. For example, hydrological models have been used to gap fill missing data from streamflow monitoring stations [8]. However, this approach does not scale well as physics-based models can be resource intensive to calibrate and do not yet effectively model many phenomena at resolutions that match sensor data.

Data gaps in sensor-collected datasets can sometimes be very large (i.e., spanning multiple days). For example, the water quality sensor data introduced in Section V Experiment Results and Analysis has many large data gaps due to power losses, which prevented the nearly continuous sampling (every 15 min) sensors from collecting data until power was reestablished at the site. The consequent large data gap can make data imputation challenging. In fact, the accuracy of data imputation methods can drop very fast as the data gap increases [9], [10]. The data imputation problem becomes even more challenging if it is a multivariate data imputation problem (i.e., multiple variables have missing data) compared to a univariate data imputation problem (i.e., only one variable has missing data). To the best of our knowledge there has been little research conducted on the question **"How to predict a continuous large data gap for a multivariate time series dataset using data from the same sensor station?"**. In this paper, we propose a data imputation framework to address

Rui Wu is with the Department of Computer Science, East Carolina University, East 10th Street and Founders Drive Greenville, NC 27858 USA. (e-mail: WUR18@ECU.EDU).

Scott D. Hamshaw is with the Department of Civil & Environmental Engineering, University of Vermont, Burlington, VT 05405 USA. (e-mail: Scott.Hamshaw@uvm.edu).

Lei Yang is with the Department of Computer Science & Engineering, University of Nevada, Reno, 1664 N Virginia St, Reno, NV 89557 USA. (e-mail: leiy@unr.edu).

Dustin W. Kincaid is with Vermont EPSCoR and the Gund Institute for Environment, University of Vermont, Burlington, VT 05405 USA. (e-mail: dustinkincaid@gmail.com).

Randall Etheridge is with the Department of Engineering and Center for Sustainable Energy and Environmental Engineering, East Carolina University, East 10th Street and Founders Drive Greenville, NC 27858 USA. (e-mail: etheridgej15@ecu.edu).

is with the Department of Computer Science and Engineering, California State University, San Bernardino, 5500 University Pkwy, San Bernardino, CA 92407 USA. (e-mail: amir.ghasemkhani@csusb.edu).

this research question by leveraging the inherent correlation between the missing data and the observed data.

Specifically, data interdependencies need to be built between missing data and the limited information collected from one sensor station. One possible method for building the interdependencies is regression machine learning models [11], [12]. Regression machine learning models can predict missing data by mathematically learning the relationship between target variables with missing values and other variables. The relationship can be expressed with different structures, such as trees, equations, and neural networks. These structures should be dynamically created based on historical data. However, regression models are trained to predict missing data of one variable based on other variables. As such, if multiple variables have missing data, the regression model cannot be directly applied to solve the multivariate data imputation problem. To address this issue, we propose using iterative data imputation, such as the multiple imputation by chained equations (MICE) method [13], to predict missing values. The MICE algorithm leverages a chain of regression models and allows the use of previously imputed values to predict subsequent variables. This means MICE can be applied to solve multivariate data imputation problems [14]. However, MICE is designed for general data imputation and may not consider the temporal connections between each data record. To help MICE discover the intrinsic correlation in the dataset, we propose the use of a reshape operation, which builds upon the sliding window and lagged correlation approaches [15] to improve MICE performance. The reshape operation can potentially enhance temporally repeating patterns within time series data and improve the correlation between missing and observed values.

In this work, we mainly employ two strategies to improve the performance of the MICE method:

- Reshape: this is a data organization modification operation which is helpful to strengthen data interdependencies by leveraging temporal information [16]. Data records with different timestamps can be combined as a new data record. This operation can potentially build connections between data collected before or after the gap with missing data because they might have a similar temporal pattern. For example, daily water temperatures oscillate similarly each day. Sensor collected data can be combined based on the time cycle to enhance interdependencies between missing temperature values and observed temperature values. The idea is commonly used in autoregressive machine learning models for time series data prediction [17], [18]. In this paper, we explore how to reshape a multivariate dataset with large data gaps to enhance the interdependencies between missing and observed data.
- Extreme value separation: extreme values in a time series are extremely small or large values that infrequently occur [19]. Conversely, normal values are commonly observed according to the data distribution. For example, values between 10 and 90 percentiles, i.e., normal values, have much higher probabilities compared to the values below 10 percentile or above 90 percentile, i.e., extreme values. These extreme values can negatively impact data prediction [20]. One possible method to mitigate the impacts of extreme values is to separate the original data into normal and extreme values and separately implement data imputation models on each dataset. However, splitting time-series sensor data without losing temporal information from the data is challenging. We introduce a method for extreme value separation in Section III B.

Our proposed framework, i.e., Large Gaps Data Imputation (LGDI), for multivariate sensor data contributes an advancement in available methods for gap filling missing data in datasets where gaps can be large (i.e., 20% missing values). Our framework shows how to integrate reshape and extreme value separation operations with MICE algorithm to enhance the data interdependencies and split data into normal and extreme categories. We also illustrate possible impacts of different reshape methods (i.e., combining records with small vs. large time intervals) and demonstrate which operations should be applied first, reshape or extreme value separations. Additionally, we test the impacts of data gap sizes on the accuracy of LGDI against selected popular data imputation algorithms (i.e., GAIN [21], MRNN [22], and MICE [13]). The LGDI source code is available at [23].

The proposed method is applicable to time series datasets including extreme and normal values with repeating temporal patterns and large data gaps. In this work, we demonstrate the application of LGDI using data from stream water quality monitoring stations equipped with continuously observing sensors measuring meteorological and hydrological parameters. The remaining sections of this paper review related work on machine learning for data imputation (Section II), present connections between the reshape operation, sliding window, and lagged data records (Section III), present our proposed LGDI framework (Section IV), present the experiment results (Section V), and discuss our conclusions and recommendations for future work (Section VI).

II. Related Work

Machine learning models are commonly used for time-series data imputation. Data imputation can be treated as either a supervised learning or an unsupervised learning task. For the supervised learning problem, variates with missing values will be treated as labels (i.e., what we want to predict). Neural networks can be constructed based on a back propagation algorithm to estimate the values of records with missing values [24]. If only one variate has missing values, the data imputation is a basic application of a regression machine learning model that relies on other variates for prediction. However, if multiple variates have missing values, a regression machine learning model cannot be directly applied. Advanced neural networks, such as generative adversarial networks (GANs) and recurrent neural networks [22], [25], [26] offer a more capable solution to missing value imputation, including the multivariate case. For example, Yoon et al. [21] proposed the Generative Adversarial Imputation Nets (GAIN) framework to address this issue. Missing values are estimated using a GAN method [27], where one neural network (the generator) uses a noise variable to generate predictions and another neural

network (the discriminator) determines whether the predictions are from the true data distribution or not. After both neural networks are trained, the generator can generate accurate estimations for missing values. Multi-Directional Recurrent Neural Networks (MRNN) is based on the deep learning architecture including an interpolation block and an imputation block to estimate missing values [22]. This method leverages the temporal information within the data and both correlation within one variable and across other variables. These two methods are very promising and were compared to LGDI in this work.

Unsupervised learning data imputation methods, such as k-means and neural network clustering algorithms, have also been applied to missing value imputation. Different from supervised learning, unsupervised learning does not have a training phase and does not use label information (i.e., observed values in the target variate with missing data). Missing values in a time series can be estimated using neighboring data records. For example, spatiotemporal self-organizing maps (SOMs) can leverage temporal correlations within data to find neighbor records [28]. This method can be improved when missing values are replaced by weighted nearest neighbors' mean [29]. To leverage both spatial and temporal information, Gowgi and his colleagues propose the use of spatiotemporal memories to learn temporal dynamics and to calculate the closeness according to a spatiotemporal metric [30]. Thus, we use temporal information and correlation to combine data records with their neighbors. Spatial information is not used because we assume data is only available from one sensor site (see Introduction).

Few data imputation studies specifically address larger data gaps [31], [32]. However, a few studies discuss how to adjust their algorithms for large missing gaps. Two possible relations that can be leveraged to tackle the large missing gap challenge are i) correlations between variables with missing values and other variables and ii) temporal dependencies between data records with missing values and data records without missing values. If sensor collected data is organized as a matrix, each column can be values of a variable and each row can be a data record at a timestamp. "correlations" are the connections between each pair of column and "temporal dependencies" are the relations within the rows. Missing values usually are not independent. They can have close correlations with other variables collected at the same or different times. One example of an approach that leverages correlations is the ratio-based imputation algorithm to handle large gaps [31]. In this method, correlations are calculated and ranked for each pair of variables. The missing values are then estimated based on a linear model and the inputs of the linear model are selected based on the correlation rank. When there are strong correlations between missing values and observed values, the proposed ratio-based imputation method is very accurate. However, the original MICE algorithm performs better than the ratio-based method when the data gap is large and variables are weakly correlated correlated [31]. For example, if Spearman correlation value is calculated and close to 0, it means two variables are not strongly correlated. With temporal dependencies, time intervals between two collected data records can be used as observation patterns and dependencies [25]. This information can help machine learning models learn connections between missing and observed values. The time intervals may not be fixed and can dynamically change. Machine learning methods, such as KNN, can be used to find these patterns and estimate missing values [33].

Our proposed data imputation framework, LGDI, is based on iterative imputation. MICE is one of the most prevalent iterative imputation methods [13]. This method initially fills missing values with temporary values using simple methods, such as the mean value of non-missing data. Subsequently, the method imputes missing values for one variable (i.e., column of data from the filled matrix) at a time by training a regression model with the remaining variables as predictive features. This step is repeated for each variable. Regression training is iterated multiple times until the predicted values converge. MICE is a powerful iteration method, because it can predict multiple missing values at the same timestamp according to the variable connections obtained from historical data. However, large data gaps can substantially reduce MICE accuracy [9], [31]. This is because the original MICE does not consider temporal dependencies. We propose to use reshape operation to solve this issue.

Reshaping the data is a potential method to enhance missing data prediction. For example, one proposed reshape method creates a matrix based on sensor data. Each row represents a record at a certain timestamp and each column includes data observations of a variable. A new record can be created by combining multiple rows. However, the reshaped matrix should avoid entire rows or columns with all missing values [16]. Thus, this method cannot be directly applied to our problem because our datasets have large data gaps where entire columns (all observations for some variables in the matrix) are missing after the reshape operation. In this paper, we describe a reshape solution that uses reshape and matrix split method to overcome this issue.

By using the reshape and extreme and normal value split operations, our proposed LGDI can perform more accurately on large missing gaps compared to existing approaches. Reshape can combine records with similar patterns to increase the correlation between missing and observed values. To treat data with different distributions uniquely, LGDI splits time series data into extreme and normal groups and uses different machine learning models for these two groups.

III. Connection between Reshape and Sliding Window

Sliding window can be used to convert a time-series supervised learning problem into a classic supervised learning problem [15]. The main idea is to use data records at previous timestamps to forecast future values, which is commonly used in autoregressive models [34]. The current sensor data can be expected to be correlated with observed data records. An autoregressive model of order p is defined as:

$$X_t = c + \sum_{i=1}^{p} \varphi_i X_{t-i} + \epsilon_t \tag{1}$$

where X_t is the time series value at timestamp t; c is a constant; φ is the parameter of an autoregressive model; and ϵ_t is the noise term. The prediction of X_t is based previous p observations. Multivariate time series sensor data with n different variables can have similar relationship as shown in the following equation:

$$X_{m,t} = c + \sum_{j=1}^{m}\sum_{i=1}^{p} \varphi_{i,j} X_{j,t-i} + \epsilon_t \tag{2}$$

where m denotes the number of variables leveraged for X_{mt} prediction at timestamp t. To generalize this idea for more complex cases, operations besides sum can be applied and Eq. (2) can be written as:

$$\begin{aligned} X_{m,t} = c + f(X_{1,t-1}, X_{1,t-2}, .., X_{1,t-p}, \\ X_{2,t-i}, ..., X_{m,t-p}) + \epsilon_t \end{aligned} \tag{3}$$

where function $f(\cdot)$ can be a regression machine learning model and takes the lags of multiple variables as features. The reshape operation creates new records and can predict missing values using the lag idea introduced in Eq. (2). The value p should be selected based on the length of repeated data patterns, which can be calculated by the autocorrelation formula [35]. For example, the Potash research site data is used in the experiment results and analysis section. Its autocorrelation values are calculated and visualized in Fig. 1b. For example, the second highest peak p value at the Potash study site occurs around 13 days (Fig. 1a), and thus p is 13 day.

The sliding window can leverage temporal information within target variable y and connections between target variable y and related features to forecast future y values. In this paper, we apply a similar idea to reshape the data organization by combining records with missing values with their neighboring records to estimate missing values from observed values. Reshape can improve the results because it can add features that have strong connections with missing values from other timestamps.

1) Reshape Method 1: Cut Along Rows: Here we discuss two options for reshaping the data to apply the sliding window concept discussed previously. The first method reshapes the data by merging data records with different timestamps. For example, a sensor dataset can be organized as the following matrix (first column is timestamp and rest of the columns are $m-1$ variables), in which each row can represent a record at timestamp n.

$$D_{n,m} = \begin{pmatrix} t_1 & a_{1,1} & a_{1,2} & \cdots & a_{1,m-1} \\ t_2 & a_{2,1} & a_{2,2} & \cdots & a_{2,m-1} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ t_n & a_{n,1} & a_{n,2} & \cdots & a_{n,m-1} \end{pmatrix}$$

Every two rows are combined into a new row, i.e., data at timestamp $t-1$ is combined with data at timestamp t and the reshaped matrix turns into:

$$D_{n/2,2m} = \begin{pmatrix} t_1 & \cdots & a_{1,m-1} & t_2 & \cdots & a_{2,m-1} \\ t_3 & \cdots & a_{3,m-1} & t_4 & \cdots & a_{4,m-1} \\ \vdots & \ddots & \vdots & \vdots & \ddots & \cdots \\ t_{n-1} & \cdots & a_{n-1,m-1} & t_n & \cdots & a_{n,m-1} \end{pmatrix}$$

If n is not an even number, the last row will be created by duplicating the record at timestamp n. If $variable_m$ (i.e., $variable$ at column m) has missing values at timestamp t, then a regression model is built as Eq. (4) shows.

$$\begin{aligned} X_{m,t} = c + f(X_{1,t-1}, X_{2,t-1}, ..., X_{m,t-1}, \\ X_{1,t}, ..., X_{m-1,t}) + \epsilon_t \end{aligned} \tag{4}$$

To generalize this reshape operation, we can choose to combine n^* data records, where n^* is a factor of n. If the original sensor data is organized as a n by m matrix (n denotes timestamp and m denotes the number of variables), then the data becomes a $\frac{n}{n^*}$ by $m * n^*$ matrix. MICE can predict each missing value by training a regression model with $m * n^* - 1$ features.

Missing data in a time series are often strongly correlated with neighboring data. Reshaping the dataset prior to MICE imputation, which does not consider temporal connection between data records, can improve imputation accuracy by enhancing temporally repeating patterns within time series data and improving the correlation between missing and observed values. However, combining more data records (i.e., large n^*) will not necessarily improve the imputation results. This is because combining multiple data records can also increase the chance of including more missing data in each combined data record (i.e., row). This can make it challenging to predict missing values with iterative imputation methods. Additionally, combining multiple data records can cause a curse of dimensionality issue [36]. Because MICE needs to predict the missing data variable with other variables as features, the more data records combined means that more features will be in each combined data record. Thus, it is necessary to conduct experiments to determine a reasonable n^* to balance the benefits of connections with neighbors and the disadvantages of combining too many records for a specific sensor dataset with missing values. A reasonable n^* value should avoid rows and columns with too many missing values [16].

If the missing data gap is large, i.e., a variable has continuous missing data points, the Reshape Method 1 may not improve the imputation accuracy. This is because reshaping data with large data gaps can produce a combined row with many missing values of the same variable. In this scenario, the MICE regression models, which makes predictions based on predictions of the same variable multiple times, will likely perform poorly and produce inaccurate results. For example, if a_{11} and a_{21} of the original matrix $D_{n,m}$ are NAN and every two rows are combined, then the first row of the reshaped matrix $D_{n/2,m}$ will have no information on $variable_1$. It is difficult for a regression model in the MICE algorithm to predict $variable_1$ at timestamp t_1 and t_2. This issue will

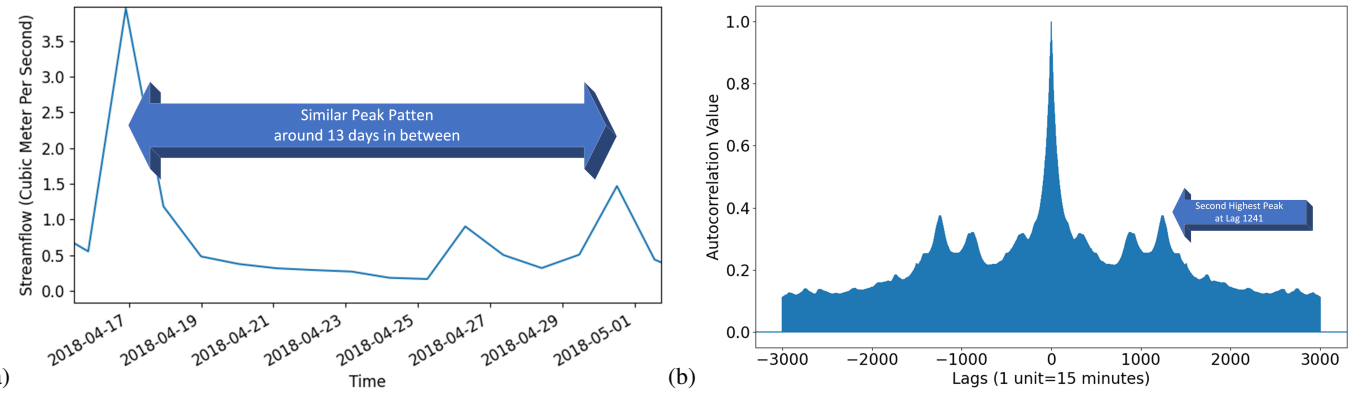

Fig. 1: (a) Example stream flow time series from Potash Brook. (b) Autocorrelation values of streamflow for the time period shown in (a). The highest autocorrelation occurs at lag 0 and the second highest at lag 1241 (13 days). A similar peak pattern is shown in (a).

become more challenging when multiple rows are combined with a variable that has many continuous missing values.

2) Reshape Method 2: Cut Along Columns: To address this large data gap challenge, we propose an alternative reshape method that builds on the approach discussed previously [16]. If the original matrix is n by m each column can be split into n'^* chunks with the same length and each chunk will be treated as a new column and combined horizontally. Then the reshaped matrix will be $\frac{n}{n'^*}$ by $m * n'^*$ (Figure 2).

Compared to the Reshape Method 1, the lags used in each row are not close. However, this can be beneficial if a variable has a large gap. In matrix $D_{n,m}$, if there is a large data gap from timestamp 1 to timestamp $n/2$ for $variable_1$ (i.e., from $a_{1,1}$ to $a_{n/2,1}$) and the Reshape Method 1 is applied to combine every two rows (n^* is 2), then there will be no information for $variable_1$ (i.e., $a_{1,1}$ to $a_{n,1}$) available for the first $n/2$ rows. In this case, the MICE data imputation might perform poorly. If Reshape Method 2 is applied and $n'^* = 2$ is used, then the matrix will be:

$$D'_{n/2,2m} = \begin{pmatrix} t_1 & t_{n/2+1} & a_{1,1} & \cdots & a_{n/2+1,m-1} \\ t_2 & t_{n/2+2} & a_{2,1} & \cdots & a_{n/2+2,m-1} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ t_{n/2} & t_n & a_{n/2,1} & \cdots & a_{n,m-1} \end{pmatrix}$$

Even if there is a missing gap from $a_{1,1}$ to $a_{n/2,1}$, there is a $variable_1$ observed value (i.e., from $a_{n/2+1,1}$ to $a_{n,1}$) in each row to predict missing $variable_1$ values. Even though the $variable_1$ lag of matrix $D'_{n/2,2m}$ is much larger compared to reshaped matrix from Reshape Method 1 $D_{n/2,2m}$ (i.e., 2 vs $n/2$), our experimental results show that the MICE algorithm can perform more accurately with the matrix $D'_{n/2,2m}$. However, Reshape Method 1 may be more appropriate for MICE if large data gaps are not a feature of the dataset, because of the shorter lags (see Section V for more details).

IV. LGDI Imputation Framework

To address the problem of imputing multivariate data with large data gaps, we propose a data imputation framework that leverages data transformation and Reshape Method 3 to improve MICE model accuracy (Figure 3). Reshape Method 1 and 2 are discussed in the previous section. To achieve more accurate data imputation results, we propose Reshape Method 3 which combines the extreme value separation with Reshape method 2. More details are introduced in Section IV-B.

A. Split Data Into Extreme and Normal Values

This step labels extreme values in the time-series data file and splits data into two groups: normal and extreme values. Because the data includes missing data, it is challenging to use machine learning-based classification algorithms to classify whether a data record is an extreme or a normal value. Statistical methods, such as extreme value theorem, can be customized based on the data characteristics, and used to label the extreme values. One proposed approach uses a time-series extreme value separation algorithm [37] based on Extreme Value Theory [38], [39]. The key idea is to dynamically update a threshold to split data according to the data distribution within a period of time.

Reshaping the data must be done prior to splitting the data into extreme and normal values, otherwise the important temporal information will be mixed up. For example, if the original matrix is 10×3. t_1, t_3, t_6, t_{10} are extreme values and $t_2, t_4, t_5, t_7, t_8, t_9$ are normal values as shown in the following two matrices:

$$D_{extreme} = \begin{pmatrix} t_1 & a_{1,1} & a_{1,2} \\ t_3 & a_{3,1} & a_{3,2} \\ t_6 & a_{6,1} & a_{6,2} \\ t_{10} & a_{10,1} & a_{10,2} \end{pmatrix}$$

$$D_{normal} = \begin{pmatrix} t_2 & a_{2,1} & a_{2,2} \\ t_4 & a_{4,1} & a_{4,2} \\ t_5 & a_{5,1} & a_{5,2} \\ t_7 & a_{7,1} & a_{7,2} \\ t_8 & a_{8,1} & a_{8,2} \\ t_9 & a_{9,1} & a_{9,2} \end{pmatrix}$$

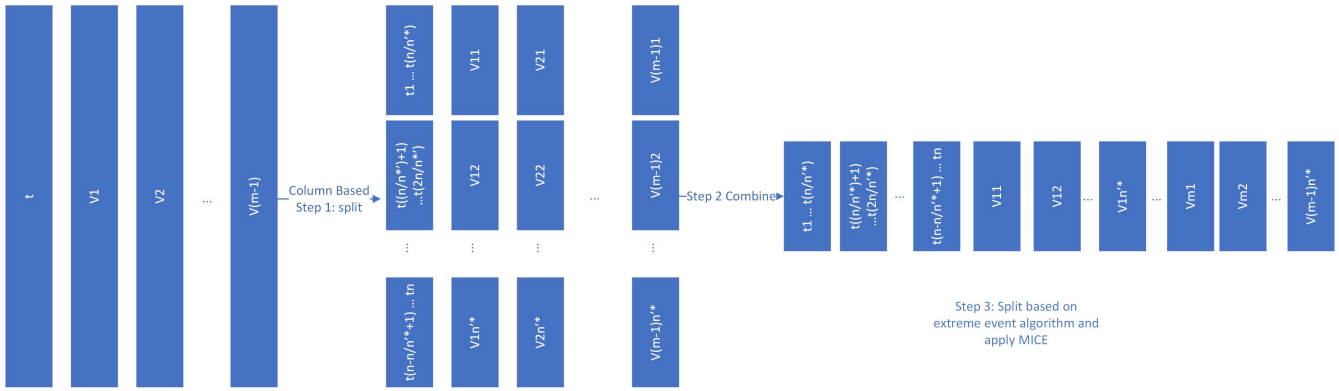

Fig. 2: In the Reshape Method 2, the original matrix is split into columns. Each column is subsequently cut into equal n'^* chunks including $\frac{n}{n'^*}$ records. The column chunks are then combined horizontally. The subsequent MICE data imputation is applied to the reshaped matrix. t denotes the timestamp column and $V1...V(m-1)$ denote $variable_1$ to $variable_{m-1}$

If Reshape Method 1 is used to combine every two rows, the two matrices become:

$$D_{extreme-reshape} = \begin{pmatrix} t_1 & a_{1,1} & a_{1,2} & t_3 & a_{3,1} & a_{3,2} \\ t_6 & a_{6,1} & a_{6,2} & t_{10} & a_{10,1} & a_{10,2} \end{pmatrix}$$

$$D_{normal-reshape} = \begin{pmatrix} t_2 & a_{2,1} & a_{2,2} & t_4 & a_{4,1} & a_{4,2} \\ t_5 & a_{5,1} & a_{5,2} & t_7 & a_{7,1} & a_{7,2} \\ t_8 & a_{8,1} & a_{8,2} & t_9 & a_{9,1} & a_{9,2} \end{pmatrix}$$

The time intervals between each combined row are not consistent because data is split first and then reshaped (same for other reshape methods introduced in this paper). In $D_{extreme-reshape}$, the time interval of the first row is $t_3 - t_1$ which is different from the second row time interval $t_6 - t_{10}$. The same problem exists in $D_{normal-reshape}$. It is challenging for regression models used in MICE to find the data patterns if the time interval is not consistent. To avoid this issue, reshaping the data must be done prior to splitting the data into extreme and normal values Figure 3.

Before we continue, we need to introduce a rule to determine if a combined record is treated as extreme. One potential rule is the "OR" rule in which a combined row is treated as an extreme if the row includes an extreme value. For example, t_1 and t_2 are combined. Because t_1 is an extreme value, the combined row is considered extreme. If Reshape Method 1 is used to combine every two rows (i.e., $n^* = 2$) and the "OR" rule is applied, then the extreme and normal reshaped matrices will be:

$$D'_{extreme-reshape} = \begin{pmatrix} t_1 & a_{1,1} & a_{1,2} & t_2 & a_{2,1} & a_{2,2} \\ t_3 & a_{3,1} & a_{3,2} & t_4 & a_{4,1} & a_{4,2} \\ t_5 & a_{5,1} & a_{5,2} & t_6 & a_{6,1} & a_{6,2} \\ t_9 & a_{9,1} & a_{9,2} & t_{10} & a_{10,1} & a_{10,2} \end{pmatrix}$$

$$D'_{normal-reshape} = \begin{pmatrix} t_7 & a_{7,1} & a_{7,2} & t_8 & a_{8,1} & a_{8,2} \end{pmatrix}$$

Because the matrix is reshaped and then split, the time intervals between each row are the same, i.e., n^* for Reshape Method 1 and for n/n^* Reshape Method 2.

B. Reshape Method 3: Reshape and Split Based on Lags

However, the "OR" rule can also cause problems. If most combined rows have an extreme value, the "OR" rule can cause an imbalanced issue where the reshaped extreme matrix is much larger than the normal matrix. For example, $D'_{extreme-reshape}$ has many more rows than $D'_{normal-reshape}$. MICE may not have enough data to predict missing values if a matrix has a small number of rows. To address this issue, we propose Reshape Method 3 as shown in Figure 4.

A chosen extreme value detection algorithm should be able to classify enough records (i.e., rows) from the original matrix as normal and extreme values for MICE data imputation and the reshape method should not change the ratio between extreme values and normal values to avoid producing imbalanced extreme and normal matrices. To achieve these goals, we propose Algorithm 1 to reshape and split the sensor data.

Algorithm 1 Reshape Method 3 used by LGDI

Input: D
Output: N **and** E
$Flag \leftarrow SPOT(D)$
$n'^* \leftarrow Autocorrelation(D)$
$D \leftarrow Reshape\_Method\_2(D, n'^*)$
$DL \leftarrow tile(D, (n'^*, 1))$
$DL \leftarrow$ remove $\frac{n}{n'^*}$ duplicates from DL
$DN \leftarrow hstack(D, DL)$
$N \leftarrow []$ and $E \leftarrow []$
$row\_index \leftarrow 0$
while $row\_index < n$ **do**
  if $Flag[row\_index] = 0$ **then**
    $N.append(DN[row\_index])$
  else if $Flag[row\_index] = 1$ **then**
    $E.append(DN[row\_index])$
  end if
  $row\_index \leftarrow row\_index + 1$
end while
Return N and E

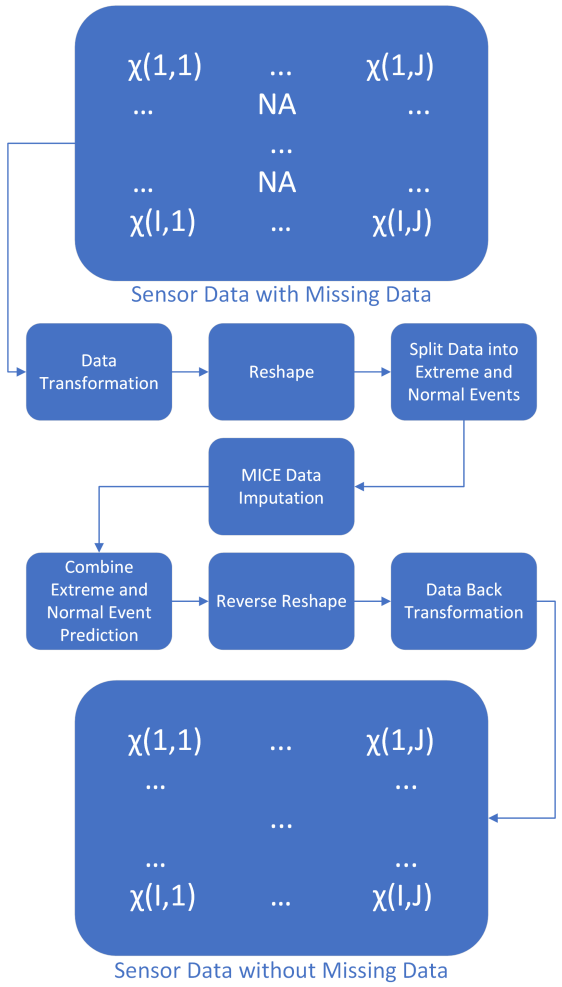

Fig. 3: The Large Data Gap Imputation (LGDI) Framework, which uses reshape, extreme value separation, and data transformation operations, can enhance correlations between missing and observed data.

Here N denotes $D_{normal-reshape}$ and E denotes $D_{extreme-reshape}$. The algorithm takes the original n by m matrix D as input and returns normal value and extreme value matrices. In this algorithm, compare to Fig. 4,

- Line 3 is Step 1. An extreme value flag vector should be created to mark if the current row of the original matrix is considered extreme. SPOT is a threshold-based algorithm and can label extreme values in streaming sensor data [37]. SPOT can be replaced with other similar extreme value labeling algorithms;
- Line 4 - 5 is Step 2. Reshape Method 2 introduced Section III 2 is applied to split each column of the original matrix into n'^* chunks. n'^* is calculated based on autocorrelation values. These chunks are combined horizontally to create a reshaped matrix. The reshaped matrix will be $\frac{n}{n'^*}$ by $m * n'^*$;
- Line 6 is Step 3. The reshaped matrix is copied n'^* times and combined vertically to be the initial lagged matrix. The lagged matrix will be n by $m * n'^*$;
- Line 7 is Step 4. The lagged matrix is compared with

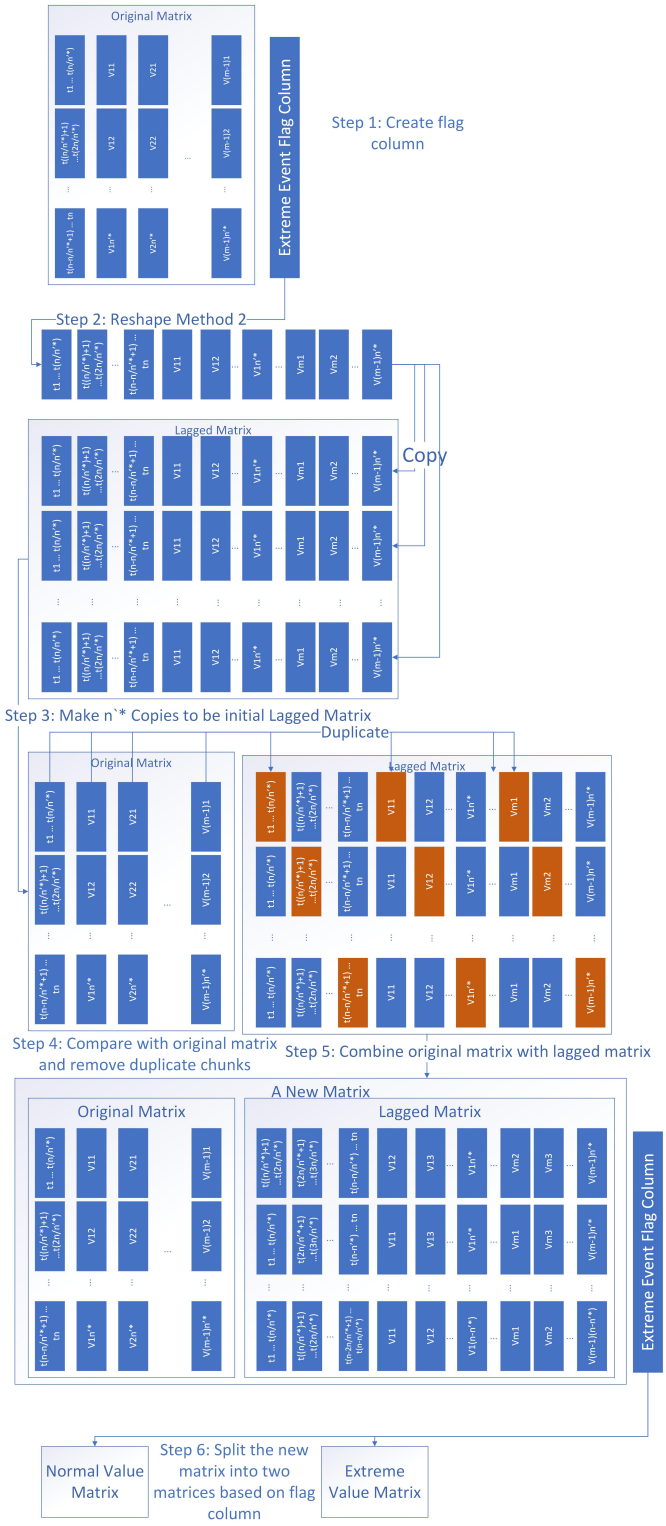

Fig. 4: In Reshape Method 3, an extreme value flag column is created to denote whether a row is considered extreme. A new matrix is created by combining the original matrix with a lagged matrix. The lagged matrix can be created by repeating Reshape Method 2 n'^* times and removing the duplicate section in the lagged matrix. The duplicated sections are highlighted as orange color and should be removed because the repeated observed and missing values can mislead machine learning imputation results.

the original matrix row by row. The repeated elements should be removed and the lagged matrix will be n by $(m * n'^* - \frac{n}{n'^*})$;
- Line 8 is Step 5. A new matrix is created by combining the original matrix with the lagged matrix. The new matrix will be n by $m + (m * n'^* - \frac{n}{n'^*})$;
- The output N and E matrices are initialized, i.e., assigned empty arrays, at line 9.
- Line 10 - 17 is Step 6. The new matrix is split into extreme and normal value matrices based on the extreme flag vector created at Line 3.

Instead of reshaping the whole matrix, Reshape Method 3 creates a new matrix by combining the original matrix with a lagged matrix. The original matrix is n by m and each column is split into n'^* chunks. The lagged matrix is $\frac{n}{n'^*}$ by $m * (n'^* - 1)$. It is $m * (n'^* - 1)$ because the duplicate chunks are removed. The key idea of Reshape Method 3 is to create a new record including records happening before or after the original record. These new records are grouped based on the extreme value concept. According to our experiment results, Reshape Method 3 can improve data imputation accuracy compared to the original MICE algorithm.

C. Combine Extreme and Normal Value Prediction and Reverse Reshape

After the data is reshaped and split into extreme and normal matrices as shown in Figure 4, two MICE models will be built to predict missing values. Then the extreme and normal matrices do not have any missing values after this step. These two matrices should be combined following the order or the "new matrix" shown in Figure 4 and the "original matrix" can be extracted from the "new matrix."

D. Data Transformation and Back Transformation

Data transformation can potentially improve the accuracy of the regression model used in MICE to predict the missing values. For example, a Box-Cox transformation can be applied to transform a variable with a non-normal distribution into a variable with a normal distribution. This is important for machine learning models because input features or activation potential of a neural network last layer are usually expected to be independent and follow normal or nearly normal distribution [40]. Non-normal distribution data can violate this and cause negative impacts on the performance. The data back transformation formula should be applied based on the selected transformation method in the data transformation step.

V. Experiment Results

A. Experiment Data

To demonstrate the application of LGDI to a real-world data set, we present the case of gap filling missing data in environmental sensor data. The data used in this work were collected from three water quality monitoring stations in three streams in the Lake Champlain Basin of Vermont in the northeastern United States. Data were collected from March to November in 2017 and 2018. Hungerford Brook (43.8 km^2; 44.918403ºN, 73.055664ºW) is a primarily agricultural watershed. The Potash Brook watershed (18.4 km^2; 44.4443318ºN, 73.2144828ºW) is primarily characterized by urban and suburban development. Wade Brook (16.7 km^2; 44.864468ºN, 72.552904ºW) is a primarily forested headwater watershed. Each watershed site has different characteristic dynamics of streamflow and water quality and thus represents three cases where variables have different temporal dynamics and degrees of correlations between variables. We note Potash Brook, being an urban/suburban stream is generally the most challenging of the three sites for any type of predictive data analysis given the highly varied environmental processes at work in a human-modified watershed system.

The dataset includes stream discharge (streamflow), water quality, and solute parameters and meteorological parameters measured continuously every 15 min by an array of in situ sensors. In total, the experimental dataset includes 19 variables besides the timestamp. In-stream water quality measurements were measured using a YSI EXO2 water quality sonde (YSI Incorporated, Yellow Springs, Ohio, United States). Discharge data (m^3/s) were acquired from a U.S. Geological Survey gaging station. Solute concentrations were estimated using s::can spectro::lyser UV-Visible spectrophotometers [41], [42] where available (Hungerford Brook Station 04293900), or calculated from stage-discharge rating curves Meteorological parameters were measured using a HOBO RX3000 weather station (Onset Computer Corporation, Bourne, Massachusetts, United States). Meteorological data were originally recorded every 5 min, but were aggregated to 15-min intervals to harmonize data with in-stream measurements. Occasional sensor malfunctions or power loss led to irregular data gaps in all time series. More information on these catchments and data collection methods are reported in [41], [43].

This dataset is representative of the type of high-frequency water quality and quantity monitoring stations that are increasingly becoming available as part of next-generation, national environmental monitoring efforts such as the National Ecological Observatory Network (NEON) [44] and the U.S. Geological Survey Next Generation Water Observing System (NGWOS) [45].

B. Experiment Results and Analysis

Given the importance of the streamflow (discharge) parameter for environmental monitoring, we consider this our primary variable of importance for testing imputation of large gaps in this proof of concept application. Though the original sensor data set includes missing values, we pre-processed the data to remove records where streamflow or all sensor variables were missing to test the application of the LGDI framework. To simulate large gaps in the sensor data, we randomly added 30% missing values to the original sensor dataset and had the streamflow variable with a continuous 20% missing value gap (i.e., about 30 days of missing data) for all data imputation algorithms. For consistency, we used a Gradient Boosting Regressor [46] as the kernel for all the MICE models. We applied a 5-fold cross validation approach and calculated R^2 values. The hyperparameters are calibrated

with the gridsearchcv function and a 5-fold cross-validation splitting strategy.

Splitting the data into extreme and normal value regimes was done by applying a recursive digital filter technique commonly used in hydrology to split streamflow data into extreme and normal values (HydRun) [47]. Extreme values in these datasets are largely driven by storm events, which cause streamflow to increase for a period of hours to days and eventually return to lower base flows. HydRun detects these extreme values using base flow separation and recession analyses. This data splitting approach may not be appropriate for non-hydrological data, and alternative approaches should be chosen based on commonly used algorithms within a domain or field, and/or based on measures of the data distribution. The separation results for our dataset are displayed in Figure 5.

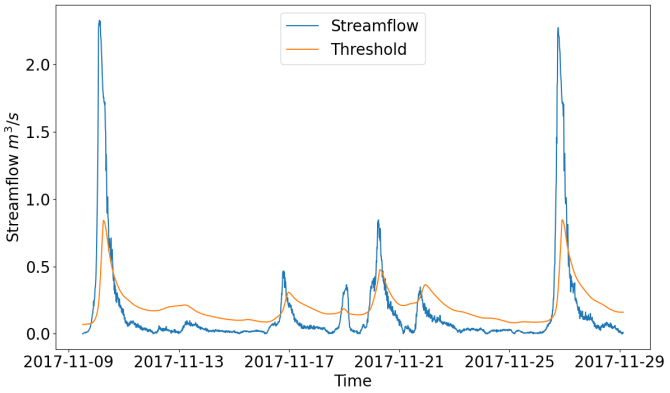

Fig. 5: An example of how observed streamflow in our experimental dataset (blue line) was split into extreme and normal values using a recursive digital filter technique (HydRun). The orange line is the threshold between normal and extreme values. All values above the threshold are treated as extreme values and all values below are treated as normal values.

Extreme values are decided by their probabilities going to happen. In Fig. 5, spike values are extreme values because most other values are close to 0. The orange threshold line is drawn based on the data distribution. All values above the orange line have a much smaller chance to happen compared to the values below the orange line.

We found that splitting the data into extreme and normal values and imputing missing values for these datasets separately as proposed in the LGDI framework improved accuracy. Table I shows the large gap missing data imputation R^2 accuracy comparisons between using a MICE model trained on the whole dataset versus a MICE model trained on just the extreme values. The latter model was more accurate (i.e., higher R^2 value) at each of the three stream monitoring sites. We surmise that the model trained on the whole dataset performed worse because it was unable to capture the periodic peak forming behavior in streamflow associated with storm-induced high flow events (Figure 5). Conversely, a model trained only on the extreme values is better able to capture these commonly occurring patterns. We also compared the overall accuracy between a single MICE model (see Table II "Original MICE" column) and predictions from normal and extreme MICE models (see Table II "Extreme and Normal Value Split" column) and again found that evaluating extreme values separately improved the accuracy of the imputation model. We note that R^2 values for Potash Brook are generally substantially lower than Hungerford and Wade Brook, which is likely associated with the urban nature of Potash Brook watershed, where streamflow is heavily altered by practices such as stormwater management, making prediction of streamflow from hydrometeorological data alone challenging.

TABLE I: Comparison of imputation model accuracy (R^2) between a MICE model trained on the whole time-series dataset and a MICE model trained only on the extreme values of the dataset.

Site	Original MICE	**Extreme Value MICE**
Wade	0.4642	0.5847
Hungerford	0.6380	0.6520
Potash	0.1387	0.3120

We compared the results of missing data imputations that used either the original MICE approach, only Reshape Method 1, only Reshape Method 2, or only Reshape Method 3. When we imputed missing data on a sensor time-series dataset without large continuous data gaps (i.e., smaller than 5% missing data gap for each variable), both reshaping methods improved the imputation model accuracy over the original MICE approach by 15 to 88%, depending on the stream monitoring site (Table III). When comparing the three reshaping methods, we found that Reshape Method 1 was more accurate than Reshape Method 2 and Reshape Method 3 at all stream monitoring sites, though increases in accuracy (around 5 to 10%) were minor relative to the improvement over not reshaping the data prior to imputation using MICE (Table III). Reshape Method 1 was more accurate than Reshape Method 2 and Reshape Method 3 because there is a lower probability that the reshaped data will have a combined row with many missing values of the same variable with smaller data gaps. Additionally, Reshape Method 1 creates new records with smaller time lags compared to Reshape Method 2 and Reshape Method 3. Conversely, when we imputed missing data on a sensor time-series dataset with large continuous data gaps, Reshape Method 2 and Reshape Method 3 were more accurate than Reshape Method 1 by 10% to 49% (Table II). This is because shorter gaps will lower the chance that a combined record has many missing values of the same variable. As such, our results demonstrate that reshaping time-series data prior to imputation of missing values using MICE improves the model accuracy substantially, and the Reshape Method 2 can further increase model accuracy on time-series data sets with large continuous data gaps. In Table III, Reshape Method 3 was more accurate compared to Reshape Method 2 (around 1% to 4%). This is because Reshape Method 2 and Reshape Method 3 used the same method to reorganize the data and Reshape Method 3 also split data into normal and extreme values, which helped machine learning models to treat data with different distributions differently.

Our experiment also demonstrated that transforming data

TABLE II: Comparison of imputation model accuracy (R^2) using different strategies to improve the accuracy of the original MICE model on a time-series dataset with large continuous missing data gaps. LGDI is the Large Data Gap Imputation framework we propose here.

Site	Original MICE	Extreme and Normal Value Split	Data Transformation	Reshape Method 1	Reshape Method 2	**LGDI**
Wade	0.4642	0.5542	0.5798	0.4179	0.5015	0.6247
Hungerford	0.6380	0.6882	0.6812	0.6113	0.6731	0.7859
Potash	0.1387	0.3165	0.3496	0.3055	0.3857	0.4507

TABLE III: Comparison of imputation model accuracy (R^2) using different methods on a time-series dataset without large continuous missing data gaps.

Site	Original MICE	**Reshape Method 1**	Reshape Method 2	Reshape Method 3
Wade	0.5291	0.9964	0.9319	0.9407
Hungerford	0.7956	0.9942	0.9128	0.9201
Potash	0.7583	0.9798	0.8917	0.9317

for normality can improve imputation model accuracy. We transformed our time-series data with large continuous data gaps prior to imputation using the Box-Cox transformation. The transformation improved the MICE model accuracy by 7 to 150% depending on the stream site (Table II). Thus, we suggest that data transformations to improve normality and stabilize variance should be explored prior to imputing missing data using MICE.

Our proposed data imputation framework, LGDI, (Figure 3) combines the strategies we tested separately here and includes data transformation(s), a reshaping method, and splitting the dataset into extreme and normal values. Currently, Reshape Method 3 is used in LGDI, though future work will improve upon this method to reduce the lagged matrix size without affecting accuracy. The LGDI framework increases the accuracy of the MICE model by 23 to 225% depending on the stream monitoring site, an improvement of 5 to 11% over the next best single strategy at each stream monitoring site (Table II, Figure 7a, and Figure 7b). For comparison purposes, we also analyzed imputed values predicted by LGDI against those generated by the GAIN and original MICE method for each site. Parts of the Hungerford site comparisons results are visualized in Figure 7a. Compared to other data imputation algorithms, LGDI generally predicted values that were less varied than MICE and also better captured peak dynamics than the MICE or GAIN method. Based on our observations, the MICE imputation results usually have more peaks for extreme data compared to LGDI. This is because the input data includes both extreme and normal events and their distributions can be different. MICE cannot account for the data distribution differences and overreacts to the data changes on the data peaks. On the other hand, LGDI splits data into extreme and normal categories, which mitigates the difficulties of extreme value imputation. The predictions of LSTM-RNN data imputation method [26] are smoother compared to LGDI. However, the differences between the estimated missing values and observed data can be large when a missing data gap size is very large. Besides these, we do note that capturing peak magnitude varied across the three sites with predicted values generally being biased low for Wade Brook and Potash Brook.

We also conducted experiments to test how the size of the data gaps affected imputation model accuracy using different data imputation methods, i.e., original MICE, our proposed LGDI, GAIN, MRNN, LSTM-RNN, and linear interpolation. We first randomly added 30% missing values and then created a continuous gap for a variable separately for all three stream monitoring datasets. Data imputation accuracy tended to decrease as a function of data gap size across all imputation methods (Figure 6a, 6b, 6c). Linear interpolation tended to perform worse than the other methods, i.e., the average R^2 value for all gap sizes and stream sites is 56% lower compared to LGDI, though accuracy was relatively stable as a function of gap size. With the exception of the large gap sizes (i.e., $>$ 30%) in the Wade, Hungerford, and Potash datasets, our LGDI framework had greater imputation accuracy than other methods we tested. Imputation accuracy using the LGDI framework was on average 7.2% greater than the next best performing method across all three stream sites. For large gap sizes (i.e., $>$ 30%), the LGDI framework outperformed the next best method by 5% to 11% depending on the stream monitoring site.

We conducted experiments to study if LGDI can be applied to other data imputation methods and MICE with traditional regression models as kernels to improve the imputation results. Wade, Hungerford, and Potash sensor datasets were used with 30% missing values overall and 20% continuous missing gap in the streamflow variable. According to the R^2 values in Table IV, LGDI is promising to improve the accuracy of other data imputation methods and MICE with traditional regression models as kernels too. For example, after applying the LGDI framework, GAIN is averagely improved 22.8%, MRNN is averagely improved 19.2%, and LSTM-RNN is averagely improved 13.6%.

VI. Conclusion and Future Work

Our experimental results demonstrate that our proposed multivariate data imputation framework, LGDI, improves the accuracy of data imputation using a MICE approach. While imputation accuracy tended to decrease with the duration of the gap, the LGDI framework had greater imputation accuracy than other imputation methods. Thus, we recommend our LGDI framework for imputing missing data on multivariate datasets that have large continuous data gaps. Our experiment results demonstrate a straightforward approach to the challenge of imputing missing data from a sensor station without relying on a neighboring station or a separate physical-based model. We anticipate the scenario we simulated of imputing one variable with a large missing gap using multivariate sensor data with random gaps is becoming increasingly common with

TABLE IV: Comparison of imputation model accuracy (R^2) with and without LGDI.

Site	Linear	KNN	MICE-SVR	MICE-Decision Tree	MICE-Random Forest	GAIN	MRNN	LSTM-RNN
Wade without LGDI	0.3626	-1.0948	-2.067	0.084	0.5249	0.4110	0.4786	0.5913
Wade LGDI	0.0804	0.0822	0.0657	0.2482	0.5398	0.6001	0.6449	0.6714
Hungerford without LGDI	0.5079	0.3764	0.3913	0.332	0.6043	0.4842	0.6908	0.7087
Hungerford LGDI	0.3219	0.4313	0.5774	0.3053	0.6239	0.5408	0.8725	0.7615
Potash without LGDI	0.2772	-0.392	-2.7949	0.1561	-0.0055	0.3189	0.4292	0.3448
Potash LGDI	0.2823	-0.3145	-0.3921	0.2943	0.4562	0.3533	0.4125	0.4133

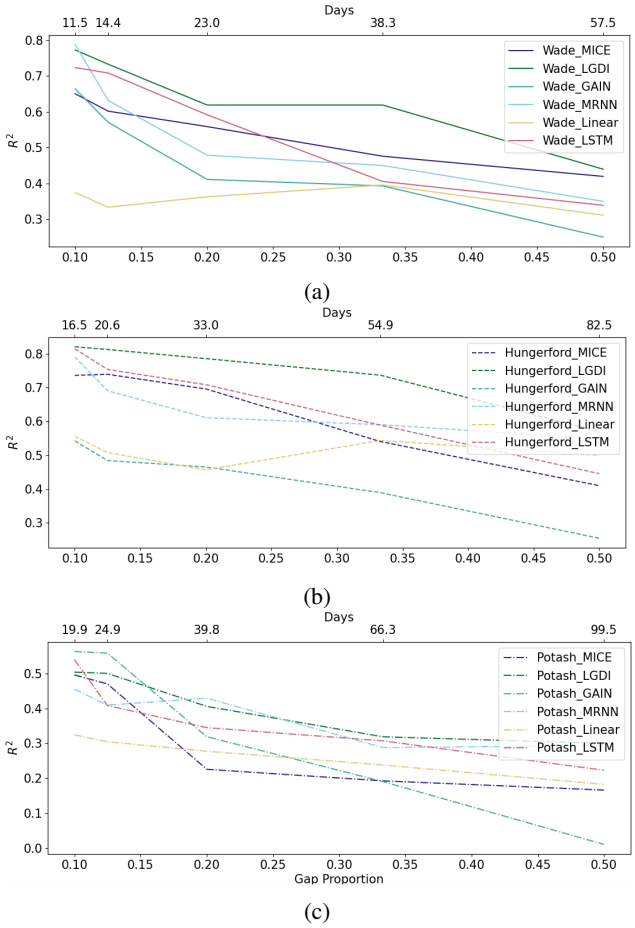

Fig. 6: Two numerical solutions: Imputation model accuracy (R^2) as a function of the data gap size across three time-series datasets from different stream monitoring sites using different data imputation methods.

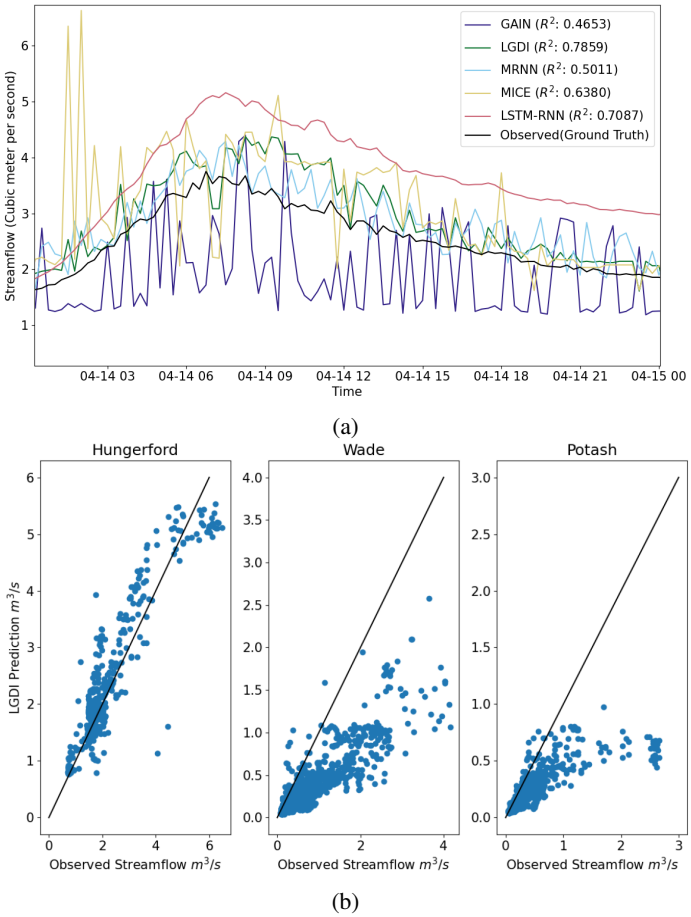

Fig. 7: (a) Data imputation predictions for MRNN, GAIN, LGDI and MICE methods vs partial observed streamflow data from Hungerford Brook (i.e., ground truth) with 30% missing values overall and 20% continuous missing gap in the streamflow variable. (b) Observed values vs LGDI predictions. The R^2 values for Hungerford, Wade, and Potash sites are $0.7859, 0.6247, 0.4507$.

multi-sensor monitoring sites used in a variety of environmental domains.

The tradeoff for improved accuracy with this proposed framework is the extra calculations and computation time required. For example, the new matrix shown in Figure 4 is much larger than the original matrix and MICE models will consume extra time for missing value estimations. Future work will improve the Reshape Method 3 to reduce the lagged matrix size without affecting accuracy.

REFERENCES

[1] A. Gelman and J. Hill, *Data analysis using regression and multilevel/hierarchical models*. Cambridge university press, 2006.

[2] C. L. Harvey, H. Dixon, and J. Hannaford, "An appraisal of the performance of data-infilling methods for application to daily mean river flow records in the uk," *Hydrology Research*, vol. 43, no. 5, pp. 618–636, 2012.

[3] A. T. Hudak, N. L. Crookston, J. S. Evans, D. E. Hall, and M. J. Falkowski, "Nearest neighbor imputation of species-level, plot-scale forest structure attributes from lidar data," *Remote Sensing of Environment*, vol. 112, no. 5, pp. 2232–2245, 2008.

[4] L. Li, Y. Li, and Z. Li, "Efficient missing data imputing for traffic flow by considering temporal and spatial dependence," *Transportation research part C: emerging technologies*, vol. 34, pp. 108–120, 2013.

[5] L. Li, J. Zhang, Y. Wang, and B. Ran, "Missing value imputation for traffic-related time series data based on a multi-view learning method," *IEEE Transactions on Intelligent Transportation Systems*, vol. 20, no. 8,

pp. 2933–2943, 2018.
[6] J. Liu, P. Musialski, P. Wonka, and J. Ye, "Tensor completion for estimating missing values in visual data," *IEEE transactions on pattern analysis and machine intelligence*, vol. 35, no. 1, pp. 208–220, 2012.
[7] S. Gandy, B. Recht, and I. Yamada, "Tensor completion and low-n-rank tensor recovery via convex optimization," *Inverse Problems*, vol. 27, no. 2, p. 025010, 2011.
[8] Y. Zhang and D. Post, "How good are hydrological models for gap-filling streamflow data?," *Hydrology and Earth System Sciences*, vol. 22, no. 8, pp. 4593–4604, 2018.
[9] T.-T.-H. Phan, A. Bigand, and É. P. Caillault, "A new fuzzy logic-based similarity measure applied to large gap imputation for uncorrelated multivariate time series," *Applied Computational Intelligence and Soft Computing*, vol. 2018, 2018.
[10] K. Kleinke, "Multiple imputation by predictive mean matching when sample size is small," *Methodology*, 2018.
[11] F. Arteaga and A. Ferrer, "Framework for regression-based missing data imputation methods in on-line mspc," *Journal of Chemometrics: A Journal of the Chemometrics Society*, vol. 19, no. 8, pp. 439–447, 2005.
[12] L. F. Burgette and J. P. Reiter, "Multiple imputation for missing data via sequential regression trees," *American journal of epidemiology*, vol. 172, no. 9, pp. 1070–1076, 2010.
[13] S. v. Buuren and K. Groothuis-Oudshoorn, "mice: Multivariate imputation by chained equations in r," *Journal of statistical software*, pp. 1–68, 2010.
[14] Z. Zhang, "Multiple imputation with multivariate imputation by chained equation (mice) package," *Annals of translational medicine*, vol. 4, no. 2, 2016.
[15] T. G. Dietterich, "Machine learning for sequential data: A review," in *Joint IAPR international workshops on statistical techniques in pattern recognition (SPR) and structural and syntactic pattern recognition (SSPR)*, pp. 15–30, Springer, 2002.
[16] M. Liao, D. Shi, Z. Yu, Z. Yi, Z. Wang, and Y. Xiang, "An alternating direction method of multipliers based approach for pmu data recovery," *IEEE Transactions on Smart Grid*, vol. 10, no. 4, pp. 4554–4565, 2018.
[17] M. Gan, Y. Cheng, K. Liu, and G.-l. Zhang, "Seasonal and trend time series forecasting based on a quasi-linear autoregressive model," *Applied Soft Computing*, vol. 24, pp. 13–18, 2014.
[18] M. C. Medeiros and Á. Veiga, "A hybrid linear-neural model for time series forecasting," *IEEE Transactions on Neural Networks*, vol. 11, no. 6, pp. 1402–1412, 2000.
[19] H. Kantz, E. G. Altmann, S. Hallerberg, D. Holstein, and A. Riegert, "Dynamical interpretation of extreme events: predictability and predictions," in *Extreme events in nature and society*, pp. 69–93, Springer, 2006.
[20] D. Ding, M. Zhang, X. Pan, M. Yang, and X. He, "Modeling extreme events in time series prediction," in *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pp. 1114–1122, 2019.
[21] J. Yoon, J. Jordon, and M. Schaar, "Gain: Missing data imputation using generative adversarial nets," in *International Conference on Machine Learning*, pp. 5689–5698, PMLR, 2018.
[22] J. Yoon, W. R. Zame, and M. van der Schaar, "Estimating missing data in temporal data streams using multi-directional recurrent neural networks," *IEEE Transactions on Biomedical Engineering*, vol. 66, no. 5, pp. 1477–1490, 2018.
[23] R. Wu, "LGDI." https://github.com/ruiwu1990/LGDI, 2022.
[24] A. Gupta and M. S. Lam, "Estimating missing values using neural networks," *Journal of the Operational Research Society*, vol. 47, no. 2, pp. 229–238, 1996.
[25] Z. Che, S. Purushotham, K. Cho, D. Sontag, and Y. Liu, "Recurrent neural networks for multivariate time series with missing values," *Scientific reports*, vol. 8, no. 1, pp. 1–12, 2018.
[26] H. Yuan, G. Xu, Z. Yao, J. Jia, and Y. Zhang, "Imputation of missing data in time series for air pollutants using long short-term memory recurrent neural networks," in *Proceedings of the 2018 ACM International Joint Conference and 2018 International Symposium on Pervasive and Ubiquitous Computing and Wearable Computers*, pp. 1293–1300, 2018.
[27] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial networks," *arXiv preprint arXiv:1406.2661*, 2014.
[28] F. Fessant and S. Midenet, "Self-organising map for data imputation and correction in surveys," *Neural Computing & Applications*, vol. 10, no. 4, pp. 300–310, 2002.
[29] Z.-G. Liu, Q. Pan, G. Mercier, and J. Dezert, "A new incomplete pattern classification method based on evidential reasoning," *IEEE transactions on cybernetics*, vol. 45, no. 4, pp. 635–646, 2014.
[30] P. Gowgi, A. Machireddy, and S. S. Garani, "Spatiotemporal memories for missing samples reconstruction," *IEEE Transactions on Neural Networks and Learning Systems*, 2021.
[31] D. Adhikari, W. Jiang, and J. Zhan, "Imputation using information fusion technique for sensor generated incomplete data with high missing gap," *Microprocessors and Microsystems*, p. 103636, 2021.
[32] A. Alamoodi, B. Zaidan, A. Zaidan, O. Albahri, J. Chen, M. Chyad, S. Garfan, and A. Aleesa, "Machine learning-based imputation soft computing approach for large missing scale and non-reference data imputation," *Chaos, Solitons & Fractals*, vol. 151, p. 111236, 2021.
[33] S. Oehmcke, O. Zielinski, and O. Kramer, "knn ensembles with penalized dtw for multivariate time series imputation," in *2016 International Joint Conference on Neural Networks (IJCNN)*, pp. 2774–2781, IEEE, 2016.
[34] T. C. Mills and T. C. Mills, *Time series techniques for economists*. Cambridge University Press, 1991.
[35] P. Legendre, "Spatial autocorrelation: trouble or new paradigm?," *Ecology*, vol. 74, no. 6, pp. 1659–1673, 1993.
[36] M. Verleysen and D. François, "The curse of dimensionality in data mining and time series prediction," in *International work-conference on artificial neural networks*, pp. 758–770, Springer, 2005.
[37] A. Siffer, P.-A. Fouque, A. Termier, and C. Largouet, "Anomaly detection in streams with extreme value theory," in *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 1067–1075, 2017.
[38] A. A. Balkema and L. De Haan, "Residual life time at great age," *The Annals of probability*, pp. 792–804, 1974.
[39] J. Pickands III *et al.*, "Statistical inference using extreme order statistics," *the Annals of Statistics*, vol. 3, no. 1, pp. 119–131, 1975.
[40] C. M. Bishop, *Pattern recognition and machine learning*. springer, 2006.
[41] M. C. Vaughan, W. B. Bowden, J. B. Shanley, A. Vermilyea, R. Sleeper, A. J. Gold, S. M. Pradhanang, S. P. Inamdar, D. F. Levia, A. S. Andres, *et al.*, "High-frequency dissolved organic carbon and nitrate measurements reveal differences in storm hysteresis and loading in relation to land cover and seasonality," *Water Resources Research*, vol. 53, no. 7, pp. 5345–5363, 2017.
[42] M. C. Vaughan, W. B. Bowden, J. B. Shanley, A. Vermilyea, B. Wemple, and A. W. Schroth, "Using in situ uv-visible spectrophotometer sensors to quantify riverine phosphorus partitioning and concentration at a high frequency," *Limnology and Oceanography: Methods*, vol. 16, no. 12, pp. 840–855, 2018.
[43] D. W. Kincaid, E. C. Seybold, E. C. Adair, W. B. Bowden, J. N. Perdrial, M. C. Vaughan, and A. W. Schroth, "Land use and season influence event-scale nitrate and soluble reactive phosphorus exports and export stoichiometry from headwater catchments," *Water Resources Research*, vol. 56, no. 10, p. e2020WR027361, 2020.
[44] K. J. Goodman, S. M. Parker, J. W. Edmonds, and L. H. Zeglin, "Expanding the scale of aquatic sciences: the role of the national ecological observatory network (neon)," *Freshwater Science*, vol. 34, no. 1, pp. 377–385, 2015.
[45] S. M. Eberts, C. R. Wagner, and M. D. Woodside, "Water priorities for the nation—the us geological survey next generation water observing system," tech. rep., US Geological Survey, 2019.
[46] L. Mason, J. Baxter, P. L. Bartlett, and M. R. Frean, "Boosting algorithms as gradient descent," in *Advances in neural information processing systems*, pp. 512–518, 2000.
[47] W. Tang and S. K. Carey, "Hyd r un: a matlab toolbox for rainfall–runoff analysis," *Hydrological Processes*, vol. 31, no. 15, pp. 2670–2682, 2017.

VII. Acknowledgements

We thank the Vermont Established Program to Stimulate Competitive Research (VT EPSCoR) for sharing the experimental data, specifically, Andrew Schroth and Carol Adair. Kincaid and Hamshaw were supported by the National Science Foundation under VT EPSCoR grant OIA-1556770. Data collection was further supported by the VT EPSCoR grant EPS-IIA 1330446. The data analysis research work is supported by the National Science Foundation under grant numbers IUSE/PFE:RED award #1730568.