A Missing Type-Aware Adaptive Interpolation Framework for Sensor Data

Lingqiang Chen, Guanghui Li, Guangyan Huang, *Member, IEEE*, and Pei Shi

***Abstract*— Data missing problems often occur on the Internet-of-Things domains. This article proposes a missing type-aware interpolation framework (IMA) for data loss problems in city-wide environmental monitoring systems that contain many scattered stations. To interpolate data as accurately as possible, IMA considers three aspects of information, i.e., spatiotemporal, all attributes of one measurement, and all values and accordingly develop three methods to estimate the missing data. First, we develop an improved multiviewer method, which uses the spatiotemporal correlation of data from neighbor stations to estimate random missing values. Second, we propose a new multi-eXtreme Gradient Boosting (multi-XGBoost) method that uses the values of the co-occurring and correlated correct attributes to predict the value of the missing attribute. Third, we take advantage of matrix factorization to estimate the missing parts if the data of the interpolation matrix are not all missing. To avoid the influence of uncorrelated data, IMA calculates Pearson's correlation coefficient between data of each station and uses those data from its top k highest correlation neighbors to form an interpolation matrix. Furthermore, due to the complexity of missing cases, IMA uses confidence levels in each of the three data prediction methods. For example, if the multiviewer method fails, IMA weights all valid results with confidence levels. We conduct our experiments on two real-world datasets from air quality monitoring stations in Beijing. Both datasets contain numerous missing measurements. Experimental results show that IMA outperforms other counterpart methods in interpolating the missing measurements, in terms of accuracy and effectiveness. Compared with the most related method, IMA improves the interpolation accuracy from 0.818 to 0.849 in a small dataset and from 0.214 to 0.759 in a large one.**

***Index Terms*— Data interpolation, matrix factorization, missing type-aware method, multi-eXtreme Gradient Boosting (multi-XGBoost), spatiotemporal correlation.**

I. Introduction

With the rapid development of the Internet of Things (IoT), massive data are generated every day and everywhere. There are many types of anomalies in streaming sensor data for various reasons, among which outliers, noise, and missing data errors are common [1]. To improve the quality of data, many anomaly detection methods [2], [3] are proposed for detecting sensor faults. In [4], an advanced distributed tensor-train decomposition method is proposed for processing industrial IoT big data that contain noise and redundancies.

Although these methods own a high accuracy to pinpoint different anomalies, few have been done for correcting the contaminated data. Besides, the missing data problems are often occurred for various reasons, such as communicational interference and sensor faults [5], [6]. However, data integrity is vital for some data-driven applications. For example, most machine learning models require full input vectors [7]. Meanwhile, some data, such as satellite remote sensing data [8], have a high cost for resampling. Thus, an effective data interpolation method is needed for improving the quality of sensor data.

In the literature, the incomplete data problem data can be divided into the following types: missing completely at random (MCAR), missing at random (MAR), and missing not at random (MNAR) [9]. As suggested in [10], MCAR and MAR data could be modeled effectively using univariate methods due to their short length. Nevertheless, MNAR data are more complex to estimate because they last for an extended period, and no local temporal information is available in this case. In practice, most of the missing data may be attributed to MAR [11], which are generated by isolated and independent random events [12], while data in MNAR state always present as block data missing, which are challenging to interpolate.

In recent years, diverse data interpolation methods using different principles have been used to estimate the missing values, such as association rule mining [13], [14], clustering [15], [16], K-nearest neighbor [17], [18], and matrix factorization [19], [20], and learning-based methods, such as CNN [5] and GAN [21]. For the nature of sensor data, the spatiotemporal correlation of data is often being applied in the methods above. Besides, some learning-based methods, such as XGBoost [22], construct complex models to mine the association between different attributes, which can also be applied for interpolation.

This article proposes a missing type-aware interpolation framework for filling the missing measurements in sensor data sampled at air quality monitoring stations. For briefness, we use the word station to represent the air quality monitoring station in Sections II–V. During the data preparation, IMA analyzes the original dataset and forms distance table (DT) and Pearson's correlation coefficient nearest neighbor table (PNT). DT records the distance between each pair of stations and is

Manuscript received April 1, 2021; revised June 2, 2021; accepted June 7, 2021. Date of publication June 16, 2021; date of current version July 6, 2021. This work was supported in part by the National Natural Science Foundation of China under Grant 62072216, in part by the Jiangsu Agriculture Science and Technology Innovation Fund under Grant CX(19)3087, and the 111 Project under Grant B12018. The Associate Editor coordinating the review process was Dr. Lorenzo Ciani. *(Corresponding author: Guanghui Li.)*

Lingqiang Chen and Guanghui Li are with the School of Artificial Intelligence and Computer Science, Jiangnan University, Wuxi 214122, China (e-mail: lqchen@stu.jiangnan.edu.cn; ghli@jiangnan.edu.cn).

Guangyan Huang is with the School of Information Technology, Deakin University, Melbourne, VIC 3125, Australia (e-mail: guangyan.huang@deakin.edu.au).

Pei Shi is with the Binjiang College, Nanjing University of Information Science and Technology, Wuxi 214105, China (e-mail: njxk_sp@sina.cn).

Digital Object Identifier 10.1109/TIM.2021.3089783

used for the distance-based method. PNT records the attribute correlation between each pair of stations and is used to find k nearest neighbors. In other words, only those data from its top k highest correlation neighbors are combined to form an interpolation matrix. In the interpolation process, IMA analyzes the type of missing data and adaptively chooses a proper method: the improved multiviewer method, the proposed multi-eXtreme Gradient Boosting (multi-XGBoost) method, or the MF method of interpolation matrix. The improved multiviewer method is one of the spatiotemporal correlation-based methods preferred when the spatiotemporal information is not missing. The proposed multi-XGBoost method is one of the attribute-based methods and mines the association between various attributes sampled at the same timestamp. Therefore, multi-XGBoost is suitable when measurements are not all empty. Different from the above two types of methods, MF is a relatively insensitive approach to missing types. By considering three aspects of information, IMA can estimate various missing types of data, except for those data that all stations lose all measurements for a long time. It is challenging to restore long-period block missing data but also meaningless. Therefore, we replace this type of data using past data. In the whole article, we use not a number (NaN) to represent missing data at certain measurement for ease of explanation.

In summary, our contributions are threefold.

1) We propose a missing type-aware interpolation framework (IMA), which considers three aspects of information, i.e., spatiotemporal, all attributes of one measurement and all values and accordingly develop three methods to estimate the missing data.
2) We apply Pearson's correlation coefficient-based K nearest neighbor mechanism to find the top k highest correlation neighbors for the target station with missing data. Then, the data from those neighbors are used for interpolation. It can reduce the computational complexity and avoid those data from stations with different states.
3) We evaluate our approach on two real-world datasets. The results demonstrate the advantages of the proposed method compared with six baseline counterpart methods. Furthermore, additional experiments show that IMA improves the prediction accuracy of a deep prediction model proposed in c10.1145/3219819.3219822.

The rest of this article is structured as follows. In Section II, the related work is reviewed and summarized. The important preliminaries are presented in Section III. Then, we detail the proposed IMA method in Section IV. Section V demonstrates the proposed method in extensive experiments. Finally, we conclude this article in Section VI.

II. Related Work

A practical problem that usually emerges in environmental monitoring systems is data missing due to sensor malfunction, network congestion, and vandalism [24]. Many existing methods have already been proposed for interpolating missing data [17], [19], [25], [26].

In tradition, some methods use simple statistics computed from the entire dataset (e.g., medians and mean) to fill those missing data [27]. However, data such as $PM_{2.5}$ fluctuate significantly from time to time according to various conditions, such as local emission, wind speed, wind direction, and secondary pollution. In addition, a direct method removes all records with NaNs to improve the data quality. It is a convenient data preprocessing method, but interpolation is more appropriate for those valuable or rare data. Traditional time series prediction methods, such as ARIMA [28], can also be applied for estimating missing data, but they only have a proper performance for data that have good periodicity properties.

In recent years, the spatiotemporal correlation of sensor data is often used to do interpolation. As suggested in [29], more recent sensor data values should have a higher contribution to the association rule, which will be used to interpolate missing measurements at a specific timestamp. In [13], spatial correlation between sensor nodes is incorporated to avoid low model accuracy caused by too much-unrelated data. Using Pearson's correlation coefficient, the method reduces complexity for the association rule mining algorithm as it only needs to search for rules from sensors with a high correlation coefficient. Similar to [13], the approach proposed in [17] takes advantage of Pearson's correlation coefficient and R2 testing in the sensor data, which uses a kd-tree structure to search for the nearest neighbors according to a weighted Euclidean metric. Then, the algorithm searches the tree to find the nearest neighbors and interpolate missing readings based on the values obtained from its neighbors. In addition, MF has also been widely used to estimate the missing values of data matrix [30], [31]. In [19], a mathematical approach recovers missing data by representing the spatiotemporal sensor data as a multidimensional tensor and applies the tensor singular value decomposition (t-SVD) to recover the missing values.

In addition, hybrid methods can also be used for interpolating missing readings. In [32], a k-means clustering and probabilistic matrix factorization (PMF) approaches are applied to recover missing values. Furthermore, the method in [26] combines empirical statistic models (inverse distance weighting [33] and simple exponential smoothing) and data-driven algorithms (user- and item-based collaborative filtering) to interpolate missing data in a multiviewer way. Besides, the fuzzy model can also be applied to address missing data issues. Considering two missing data situations (MAR and MCAR), an evolving fuzzy-rule-based model (eFGP) in [27] handles single missing values by developing reduced-term consequent polynomials. In addition, eFGP completes multiple missing values using the midpoints of time-varying granules evolved in the data space.

Inspired by the previous works, in this article, we proposed an interpolation framework to comprehensively analyze the data attributes, which contains the spatiotemporal correlation of measurements. Considering the different missing data types, we incorporate three interpolating methods in IMA: the improved multiviewer method, the proposed multi-XGBoost method, and the MF method of the interpolation matrix. Like other spatiotemporal-based methods, the improved multiviewer method can be applied to mine spatiotemporal correlation between data from different stations (one attribute). The

TABLE I
DESCRIPTION OF THE ABBREVIATED SYMBOLS

Abbreviations	Description
IMA	The proposed method
DT	Distance table that records the distance between each sensor
PNT	Pearson's correlation coefficient nearest neighbor table
MF	Matrix factorization
NaN	Not a Number
IDW	Inverse distance weighting method
SES	Simple exponential smoothing method
UCF	User-based collaborative filtering method
ICF	Item-based collaborative filtering method
n	Data size
q	Parameters number
w	Time window size
k	Top-k neighbors' size/ Decision tree size
d	Decision tree depth
m	Matrix factorization times
b	The number of bins in the histogram of LighGBM

proposed multi-XGBoost method is one of the attribute-based methods, which is used to construct complex dependence between different features sampled at one timestamp. When the above two kinds of methods are infeasible and all elements of the interpolation matrix are not NaNs, the MF method can use the product result of two random matrices to approximate the interpolation matrix (see more details in Section III-C). To interpolate the missing data as accurately as possible, IMA analyzes the missing data type and adaptively chooses an appropriate method: the improved multiviewer method for no spatiotemporal information loss, the proposed multi-XGBoost method for no attributes loss, and the MF method for no matrix loss. These three types of loss are defined in Section IV. Furthermore, we improve the multiviewer to be suitable for various data missing cases, seen details in Section IV-B2. Besides, we set the training cycle of our model as one year. In practice, we can update the model every year or half a year as routine maintenance.

III. PRELIMINARY

In this section, we introduce preliminary knowledge, including the XGBoost, the multiviewer method, and the MF. All the abbreviated symbols are listed in Table I.

A. XGBoost Method

XGBoost [22] is a gradient boosting decision tree algorithm, which has been widely used to solve many data science problems. Like the other boosting tree methods, XGBoost improves its regression accuracy by continually constructing new trees. During each construction, the new tree is used to fit the residuals of previous trees. To find an optimal alternative tree structure, XGBoost uses the Taylor expansion to simplify the objective function and applies a greedy mechanism in finding the best split. The final result is the sum of estimation from each tree.

IMA uses the XGBoost method as an attribute-based method. Considering multiple attributes missing cases, IMA builds XGBoost models for each attribute and estimates missing data iteratively. This improved XGBoost method is called multi-XGBoost (see more details in Section IV-B1).

B. Multiviewer Method

Multiviewer method [26] contains four interpolating algorithms: inverse distance weighting (IDW) method, simple exponential smoothing (SES) method, user-based collaborative filtering (UCF) method, and item-based collaborative filtering (ICF) method. Given an interpolation matrix $I = [v_{ij}]_{w\times(k+1)}$ to record recent data from target station and its k nearest neighbors, $v_{ij} \in I$ represents the jth readings in station i, and we assume that the station 1 (s_0) represents the target station and the missing data are located at middle of the time window ($\lfloor w/2 \rfloor$) at timestamp t.

IDW is used to estimate missing data in a global spatial view. As shown in (1), IDW assigns a weight to each available data of neighbor station according to their distance to the target station and gets the estimation $\hat{v}_{idw}$ by a weighted average method, where d is the distance between target station and its neighbor and α is an integer value

$$\hat{v}_{idw} = \frac{\sum_{i=1}^{k} v_{it} d_i^{-\alpha}}{\sum_{i=1}^{k} d_i^{-\alpha}}. \tag{1}$$

As an exponential moving average model, SES can interpolate the missing data in a global temporal view. In (2), SES sets an exponential weight $\beta(1-\beta)^{\lceil j-w/2 \rceil}$ to the data in the same time window, where β is a smoothing parameter with a range of (0, 1). In general, those recent data have a bigger weight than distant ones. Finally, SES gets the result $\hat{v}_{ses}$ by weighting the values in the time window

$$\hat{v}_{ses} = \frac{\sum_{j=1}^{w} v_{0j}\beta(1-\beta)^{\lceil j-w/2 \rceil}}{\sum_{j=1}^{w} \beta(1-\beta)^{\lceil j-w/2 \rceil}}. \tag{2}$$

UCF, as a data-driven method, has been widely applied in recommender systems. For the principle that similar users make similar ratings for similar items [34]. SES regards each sensor as a user and calculates the similarity of data in the s_0 and station i (s_i), according to (3), where NT is the number of timestamps that both two stations have data. Then, in (4), UCF uses the similarity as weight and calculates the final estimation $\hat{v}_{ucf}$ using its neighbors' data sampled at time t

$$\mathrm{sim}(s_i, s_0) = 1/\sqrt{\frac{\sum_{j=1}^{w}\left(v_{ij} - v_{0j}\right)^2}{NT}} \tag{3}$$

$$\hat{v}_{ucf} = \frac{\sum_{i=1}^{m} v_{it}\mathrm{sim}_i}{\sum_{i=1}^{m} \mathrm{sim}_i}. \tag{4}$$

Like UCF, ICF regards each timestamp as an item and calculates the similarity between time t and other candidate timestamps based on interpolation matrix I in (5), where NS is the number of stations with data in the two timestamps. Finally, ICF calculates the final estimation $\hat{v}_{icf}$ by weighting

the data in the current time window from s_0

$$\text{sim}(t', t) = 1/\sqrt{\frac{\sum_{i=1}^{k}(v_{it'} - v_{it})^2}{NS}} \tag{5}$$

$$\hat{v}_{icf} = \frac{\sum_{j=1}^{w} v_{0j} sim_j}{\sum_{j=1}^{w} \text{sim}_j}. \tag{6}$$

After getting four estimations, multiviewer gets the final result using a regression model, according to (7), where w_i is the learned weight for each interpolation method and b is a bias

$$\hat{v}_{mvl} = w_1\hat{v}_{idw} + w_2\hat{v}_{ses} + w_3\hat{v}_{ucf} + w_4\hat{v}_{icf} + b. \tag{7}$$

IMA uses the multiviewer method as a spatiotemporal method and improves the adaptability of the multiviewer method to various missing cases by assigning each submethod with weighted confidence (see more details in Section IV-B2).

C. Matrix Factorization Method

MF, as an effective pattern recognition technique, has been widely used in a variety of tasks [31]. Given a data matrix $A = [a_{ij}]_{m\,\text{x}\,n}$ that contains NaNs in several entries, the MF method first constructs two $U = [u_{ij}]_{m\,\text{x}\,k}$ and $V = [v_{ij}]_{k\,\text{x}\,n}$, where $k < \min(m, n)$, and initializes the weights of U and V using a random normal distribution method, i.e., $u_{ij}, v_{ij} \sim \text{N}(\mu, \sigma)$. Then, the MF method calculates the estimated matrix A' using the following equation:

$$A' = UV = \begin{bmatrix} a'_{11} & \cdots & a'_{1\ n} \\ \vdots & \ddots & \vdots \\ a'_{m1} & \cdots & a'_{mn} \end{bmatrix} = \begin{bmatrix} \sum_{l=1}^{k} u_{1\ l} v_{l1} & \cdots & \sum_{l=1}^{k} u_{1\ l} v_{ln} \\ \vdots & \ddots & \vdots \\ \sum_{l=1}^{k} u_{ml} v_{l1} & \cdots & \sum_{l=1}^{k} u_{ml} v_{ln} \end{bmatrix}. \tag{8}$$

To fit the original matrix, MF sets the loss function for weights learning as

$$\begin{aligned} L &= \|A - A'\|_F^2 + \frac{\beta}{2} J(U, V) \\ &= \|A - UV\|_F^2 + \frac{\beta}{2}\left(\|U\|^2 + \|V\|^2\right) \\ &= \sum_{i,j,a_{ij} \neq NaN} \left(a_{ij} - \sum_{l=1}^{k} u_{il} v_{lj}\right)^2 + \frac{\beta}{2}\left(\sum_{i,l} u_{il}^2 + \sum_{l,k} v_{lj}^2\right) \end{aligned} \tag{9}$$

where $\|\cdot\|_F^2$ represents the Frobenius norm, $(\beta/2)J(U, V)$ is the penalized term, and β is the penalty factor. Due to the existence of NaN values in the original matrix A, therefore, in (9), the MF method only accumulates square error on those entries with values. Then, stochastic gradient descent is used for updating weights in U and V, and each updated weight u'_{ij} and v'_{ij} can be obtained using (10)–(13), where η is the learning rate

$$\frac{\partial L}{\partial u_{iz}} = -2\sum_{j=1}^{n}\left(a_{ij} - a'_{ij}\right)\sum_{j=1}^{n} v_{zj} + \beta u_{iz} \tag{10}$$

$$\frac{\partial L}{\partial v_{zj}} = -2\sum_{i=1}^{m}\left(a_{ij} - a'_{ij}\right)\sum_{i=1}^{m} u_{iz} + \beta v_{zj} \tag{11}$$

$$\begin{aligned} u'_{iz} &= u_{ij} + \eta\frac{\partial L}{\partial u_{iz}} \\ &= u_{ij} - 2\eta\sum_{j=1}^{n}\left(a_{ij} - a'_{ij}\right)\sum_{j=1}^{n} v_{zj} + \eta\beta u_{iz} \end{aligned} \tag{12}$$

$$\begin{aligned} v'_{ij} &= v_{ij} + \eta\frac{L}{\partial v_{zj}} \\ &= v_{ij} - 2\eta\sum_{i=1}^{m}\left(a_{ij} - a'_{ij}\right)\sum_{i=1}^{m} u_{iz} + \eta\beta v_{zj}. \end{aligned} \tag{13}$$

MF repeats the iterative regression using (8)–(13) until the loss value is under a certain threshold. Eventually, we will get the estimation of missing data.

Compared with multiviewer and XGBoost methods, MF is missing-type insensitive. Therefore, when the current interpolation matrix loses much spatiotemporal information and values of all kinds of attributes are missing, IMA applies MF for recovering the data of interpolation matrix and further obtains the estimation of missing data.

IV. Missing Type-Aware Adaptive Interpolation Method

In this section, we first introduce the overview of our method and then detail IMA in three processes: data preprocessing, model preparation, and interpolation.

Fig. 1 lists the tables used in IMA. As shown in the original data table (OT), the spatiotemporal data contain both positions (latitude and longitude) and time parameters. We can use the position information to calculate the distance between two stations. To further mine the spatiotemporal correlation of data, IMA forms two tables in the data preparation process in Fig. 2. DT records the distance between any pair of stations, and PNT lists the neighbor ids of each station, which are sorted in descending order for correlation value.

Thereafter, IMA uses original data to train the proposed multi-XGBoost [22] and linear regression (LR) model of the improved multiviewer method for each station. The third part in Fig. 2 is about the interpolation process. Assume that station s_i collected a measurement $M = [m_1, \ldots, m_q]$ at time t and lost a value m_j. To estimate the missing data, IMA obtains recent measurements from k neighbors of s_i using PNT, the interval of data is from $t - \lfloor w/2 \rfloor$ to $t + \lfloor w/2 \rfloor$, and w is the time window size and forms an interpolation matrix $I = [d_{ij}]_{w\,\text{x}(k+1)}$, and therefore, the value to be estimated is $d_{\lfloor w/2 \rfloor, 0}$, i.e., $\hat{d}$. We define three data losing types: spatiotemporal information loss, attributes loss, and matrix loss. The definitions of the above three types are as follows.

Definition 1: The missing type of $\hat{d}$ is spatiotemporal information loss if both of the following conditions are satisfied.

OT:

Station name	Location	Time	Attri 1	Attri 2	...	Attri n
S1	[lon, lat]	t	*	NaN	...	*
...	...	...	...	...	...	...
S1	[lon, lat]	t+n-1	*	*	...	*
S2	[lon, lat]	t	NaN	*	...	*
...	...	...	...	...	...	...
S2	[lon, lat]	t+n-1	*	*	...	*
S3	[lon, lat]	t	*	*	...	*
...	...	...	...	...	...	*

OT: Original data Table
DT: Distance Table
PNT: Pearson-correlation-coefficient nearest Neighbor Table
TLT: Temporal data Loss Table
SLT: Spatial data Loss Table

DT:

Station name	S1	S2	S3	...
S1	0	d_{12}	d_{13}	...
S2	d_{12}	0	d_{23}	...
S3	d_{13}	d_{23}	0	...
...	...	...	...	...

PNT:

Station name	PN1	PN2	PN3	...
S1	Name 1	Name 2	Name 3	...
S2	Name 1	Name 2	Name 3	...
S3	Name 1	Name 2	Name 3	...
...	...	...	...	...

Fig. 1. Tables in IMA.

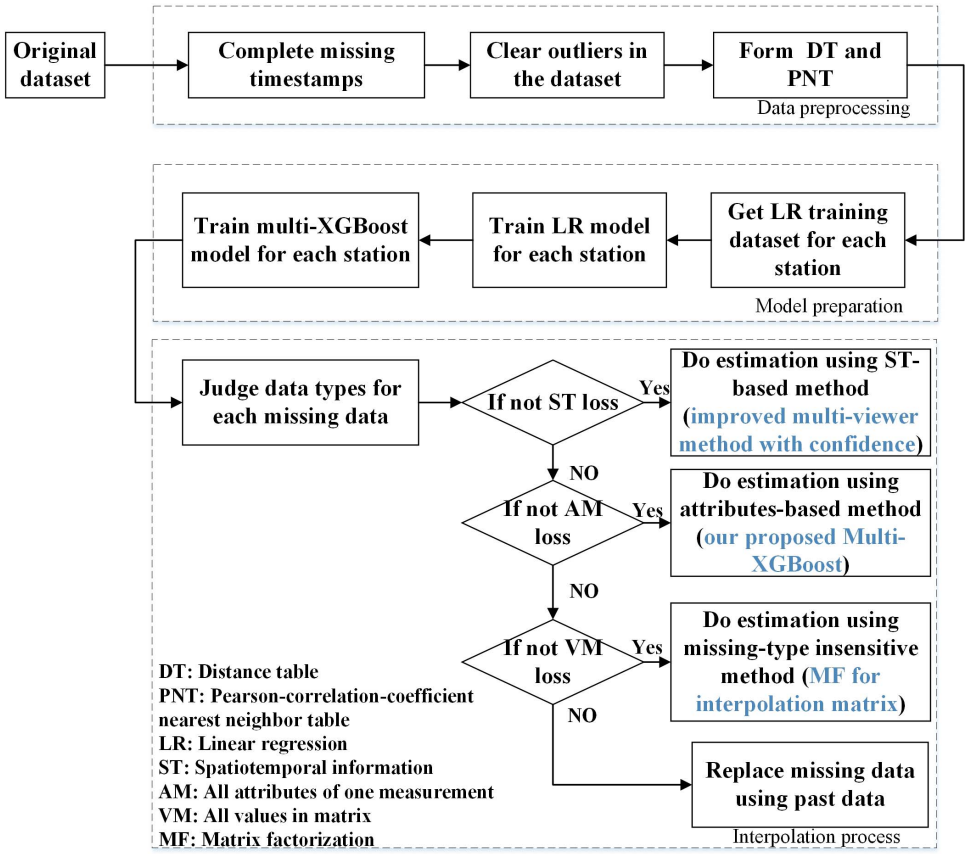


Fig. 2. Schematic of IMA.

1) If the data in the first columns of I are all NaNs, the missing data contain temporal information loss. In other words, s_i lost all measurements in this time window.
2) If the data in the middle row of I are all NaNs, the missing data contain spatial information loss. In other words, all neighbors of s_i lost jth attribute value at time t.

Definition 2: Attributes loss represents that all values in M are NaNs, i.e., $\forall\, m_j \in M, m_j = \text{NaN}$.

Definition 3: Matrix loss represents that all values in I are NaNs, i.e., $\forall\, d_{ij} \in I, d_{i,j} = \text{NaN}$.

Spatiotemporal correlation is a useful parameter of urban data and can be used in priority to estimate missing values.

However, there is a large amount of missing data, and the missing type is complex. When the two conditions of Definition 1 are satisfied, little spatiotemporal information can be used to estimate missing data. Therefore, it is proper to use other methods in different mechanisms. The attribute-based method, e.g., XGBoost, mines the relationship among different attributes sampled at the same timestamp. Thus, Definition 2 is proposed to judge whether attribute-based methods can estimate the missing data or not. If the above two data loss types are simultaneously satisfied, those methods that are not insensitive to missing data types are preferred.

A. Data Preprocessing

During the data preprocessing, the main work of IMA includes the addition of missing records, simple outlier deletion, and generation of the statistical tables in Fig. 1. We show the details in Algorithm 1.

Algorithm 1 Data Preprocessing Process

Input: Original data table: OT
Output: DT, PNT
1: Fill the missing records in OT according to timeseries
2: OT' ← Delete obvious outliers in OT
3: Form DT between each pair of stations
4: **for** attribute in all attributes **do**
5: Form PNT between each pair of station
6: **end for**

First, IMA gets a complete time series of the original dataset from start time to end time, where the interval is equal to the sampling period. Then, IMA checks the missing timestamps and inserts the empty records (1 h). Second, IMA applies a simple outlier deletion using the 90% fractile of each attribute as the threshold Θ. If the absolute difference between sampled data and its mean is larger than Θ, the data will be removed. After that, the distance between each pair of stations is calculated using the longitude and latitude of stations. Here, we apply the GeoPy module[1] for python to convert geographic information into spatial distance.

In addition, we use the original data to calculate Pearson's correlation coefficient between each pair of stations and form PNT for each attribute. Although according to the first law of geography, everything is related to everything else, but near things are more related to each other. However, data such as the concentration of $PM_{2.5}$ are influenced by many factors. As shown in Tables II–IV, we list part of PNT in our experiments and corresponding distance and Pearson's correlation coefficient value. The distance between two stations is smaller and the Pearson's correlation coefficient is higher, but not utterly inverse ratio. Therefore, IMA uses PNT as the nearest neighbor selection indicator, but not the distance.

B. Model Preparation

IMA contains three methods, among which the multi-XGBoost and the improved multiviewer methods need to pretrain their models before interpolation.

[1]https://github.com/geopy/geopy

TABLE II
EXAMPLE OF PEARSON'S CORRELATION COEFFICIENT NEAREST NEIGHBOR

Station name	Neighbor1	Neighbor2	Neighbor3	Neighbor4	...
Aotizhongxin	*Dongsi*	*Nongzhan*	*Guanyuan*	*Xizhimen*	...
Badaling	*Yanqin*	*Dingling*	*Pingchang*	*Beibuxinqu*	...
Beibuxinqu	*Pingchang*	*Wanliu*	*Zhiwuyuan*	*Mentougou*	...
Daxing	*Yizhuang*	*Yongding*	*Tiantan*	*Wanshou*	...
...	...	...	...	...	...

TABLE III
EXAMPLE OF PEARSON'S CORRELATION COEFFICIENT ($PM_{2.5}$)

Station name	Neighbor1	Neighbor2	Neighbor3	Neighbor4	...
Aotizhongxin	0.972	0.971	0.969	0.966	...
Badaling	0.887	0.852	0.805	0.788	...
Beibuxinqu	0.919	0.897	0.896	0.893	...
Daxing	0.939	0.933	0.929	0.927	...
...	...	...	...	...	...

TABLE IV
EXAMPLE OF THE NEIGHBOR DISTANCE OF SOME STATIONS (UNIT: km)

Station name	Neighbor1	Neighbor2	Neighbor3	Neighbor4	...
Aotizhongxin	6.13	7.40	7.69	**5.14**	...
Badaling	9.88	21.28	26.31	34.42	...
Beibuxinqu	14.90	14.96	**10.18**	17.97	...
Daxing	12.22	17.59	18.68	**18.34**	...
...	...	...	...	...	...

1) Training and Estimation Processes of Multi-XGBoost Model: Multi-XGBoost method is an end-to-end boosting tree model. We applied this model to learn the correlation between different attributes and build the estimation model for each attribute using other attributes. The details of the training process of the multi-XGBoost model are given as follows.

First, IMA gets the non-null datasets (D') from the original dataset to avoid NaN values' influence on the training model. Then, part of the data in D' is used for parameter optimization since multi-XGBoost contains various parameters, which results in grid search very time-consuming. Therefore, IMA constructs a parameter space of multi-XGBoost and searches the best parameters using a Bayesian optimization algorithm.[2] Then, IMA trains XGBoost models $M[n]$ for each attribute. Specifically, for obtaining the XGBoost model for attribute a_i, IMA uses all a_i data in D' as target data and feeds other attributes data to the model.

Fig. 3 shows an estimation procedure of multi-XGBoost. Common XGBoost can be applied for estimating one attribute missing with all other attributes existence. While some measurements contain multiple missing values, multi-XGBoost first fills the NaN value with the corresponding mean value. As shown in Fig. 3, IMA inserts μ_2 and μ_4 into D_2 and D_4. Consequently, IMA uses other parameters $[D_0, D_1, D_3, \mu_4]$ to predict the missing data of D_2 as E_2. Then, the estimated value will be used in the following estimation procedure. After several loops, IMA gets the final data array as $[D_0, D_1, E_2^{\#}, D_3, E_4^{\#}]$, where # is set as the size of attributes in our scheme.

[2]https://github.com/hyperopt/hyperopt

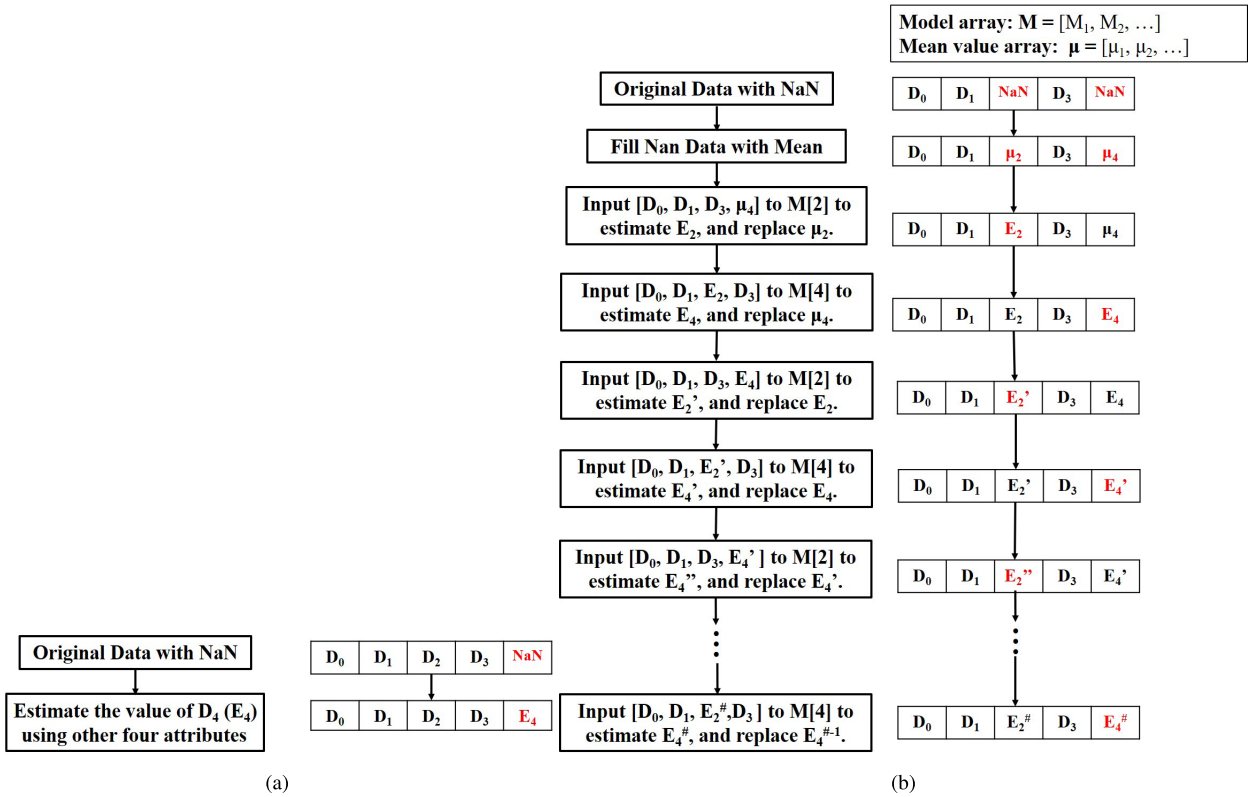


Fig. 3. Our proposed multi-XGBoost method. (a) Common XGBoost. (b) Multi-XGBoost.

Algorithm 2 Multi-XGBoost Model Learning Process

Input:
Original data: D;
Numerical attributes: $A = [a_1, a_2, \ldots, a_n]$;
Output:
Multi-XGBoost models: $M[n]$
1: Extract non-null datasets D' from D
2: Get training and testing subdataset from D' for parameter optimization
3: Construct parameter space V of XGBoost
4: Get best parameters of XGBoost using Bayesian optimization algorithm
5: **for** $i = 1 \rightarrow n$ **do**
6: Get a_i data in D' and form target datasets Y
7: Get data of other attributes in D' and form training datasets X
8: Train XGBoost model $M[i]$ using X and Y
9: **end for**

2) Training of LR Model: Data from environmental monitor systems always contain temporal and spatial correlation. Specifically, temporal correlation can be represented by similar values of data collected at close sampling timestamps on one station, while spatial correlation shows that the data simultaneously collected by adjacent stations also have high similarity. As demonstrated in Section IV-A, the spatial correlation of data between stations does not increase completely with the decrease of distance. Therefore, IMA uses PNT to find k nearest neighbor stations for the target sensor s_i. Then, s_i combines its recent data with the data from its k nearest neighbors and forms interpolation matrix $I_{w\times(k+1)}$, where w is the time window size (w is set to odd in this article) and k is the neighbor size. For convenience, we record the data of s_0 in the first column, and the missing data are $v_{0,\lfloor w/2 \rfloor}$. After constructing the interpolation matrix, IMA applies a multiviewer method [26] to estimate missing values, including IDW, SES, UCF, and ICF.

Comparing with the four methods mentioned in Preliminary (Section III-B), we have added a value Γ_i for each method in (14), (16), (18), and (20) to accommodate different missing scenarios. When the data exist NaN value, Γ_i is set as 0; otherwise, Γ_i is set as 1. For example, in (14), when v_{it} is NaN, $\Gamma_i = 0$, and otherwise, $\Gamma_i = 1$. In (18), when v_{it} or sim_i is NaN, v_{it} is NaN, $\Gamma_i = 0$, and otherwise, $\Gamma_i = 1$. Besides, a confidence value ψ is used to evaluate the reliability of the model in (15), (17), (19), and (21). Specifically, ψ trends to be a lower value when the data used for estimation contain more missing data. After getting four estimated data, IMA generates the final result, according to (22). In practice, multiviewer method may fail to make an interpolation because of the complex types of missing data. Therefore, IMA increases confidence value ψ for each estimation method. When LR fails, IMA uses confidence values to weight the valid data among four estimation results using (23)

$$\hat{v}_{idw} = \frac{\sum_{i=1}^{k} \Gamma_i v_{it} d_i^{-\alpha}}{\sum_{i=1}^{k} \Gamma_i d_i^{-\alpha}} \tag{14}$$

$$\psi_{idw} = \frac{\sum_{i=1}^{k} \Gamma_i d_i^{-\alpha}}{\sum_{i=1}^{k} d_i^{-\alpha}} \tag{15}$$

$$\hat{v}_{ses} = \frac{\sum_{j=1}^{w} \Gamma_j v_{0j} \beta (1-\beta)^{\lceil j-w/2 \rceil}}{\sum_{j=1}^{w} \Gamma_j \beta (1-\beta)^{\lceil j-w/2 \rceil}} \tag{16}$$

$$\psi_{ses} = \frac{\sum_{j=1}^{w} \Gamma_j \beta (1-\beta)^{\lceil j-w/2 \rceil}}{\sum_{j=1}^{w} \beta (1-\beta)^{\lceil j-w/2 \rceil}} \tag{17}$$

$$\hat{v}_{ucf} = \frac{\sum_{i=1}^{k} \Gamma_i v_{it} \operatorname{sim}_i}{\sum_{i=1}^{k} \Gamma_i \operatorname{sim}_i} \tag{18}$$

$$\psi_{ucf} = \frac{\sum_{i=1}^{k} 1\Gamma_i}{k} \tag{19}$$

$$\hat{v}_{icf} = \frac{\sum_{j=1}^{w} \Gamma_j v_{0j} sim_j}{\sum_{j=1}^{w} \Gamma_i \operatorname{sim}_j} \tag{20}$$

$$\psi_{icf} = \frac{\sum_{j=1}^{w} 1\Gamma_j}{w} \tag{21}$$

$$\hat{v}_{mvl} = w_1 \hat{v}_{idw} + w_2 \hat{v}_{ses} + w_3 \hat{v}_{ucf} + w_4 \hat{v}_{icf} + b \tag{22}$$

$$\hat{v}_{mvl} = \psi_{idw} \hat{v}_{idw} + \psi_{ses} \hat{v}_{ses} + \psi_{ucf} \hat{v}_{ucf} + \psi_{icf} \hat{v}_{icf}. \tag{23}$$

Algorithm 3 LR Learning Process

Input:

Original data: D; Time window: w;
Pearson's correlation coefficient nearest neighbors Table: PNT
Neighbors size: k

Output:

Linear Regression models: LR[n]

for $s_i \in$ all stations **do**
  for data $\in s_i[data]$ **do**
    Get neighbor stations of s_i using PNT
    Form interpolation matrix $I = \left[v_{ij}\right]_{w \times (k+1)}$
    $y \leftarrow v_{\lfloor w/2 \rfloor, 0}$
    Set $v_{\lfloor w/2 \rfloor, 0}$ as NaN
    Obtain estimated data $\left[\hat{v}_{gs}, \hat{v}_{gt}, \hat{v}_{ls}, \hat{v}_{lt}\right]$ using IDW, SES, UCF, and ICF
    if $\left[\hat{v}_{gs}, \hat{v}_{gt}, \hat{v}_{ls}, \hat{v}_{lt}\right]$ and y all have value **then**
      Add $\left[\hat{v}_{gs}, \hat{v}_{gt}, \hat{v}_{ls}, \hat{v}_{lt}\right]$ to training set $D_{\text{train},i}$, and add y to target set $D_{\text{target},i}$.
    end if
  end for
  Train Linear regression model LR[i] for s_i using $D_{\text{train},i}$ and $D_{\text{target},i}$.
end for

As shown in Algorithm 3, IMA builds LR models for each station. Taking one station s_i for an example, IMA first gets the neighbor information of s_i from PNT and then extracts the neighbors' data in the certain time window and forms interpolation matrix $I = [v_{ij}]_{w \times (k+1)}$.

To simulate missing data of v_{t0} for id_i, IMA records the original $v_{\lfloor w/2 \rfloor, 0}$ in the target dataset y and sets $v_{\lfloor w/2 \rfloor, 0}$ as NaN. Subsequently, IDW, SES, UCF, and ICF are applied for making estimations $[\hat{v}_{gs}, \hat{v}_{gt}, \hat{v}_{ls}, \hat{v}_{lt}]$. Since the LR model needs full data as training data, IMA only records the complete data in training set $D_{\text{train},i}$ and target set $D_{\text{target},i}$. Finally, IMA trains LR model for s_i using the data from $D_{\text{train},i}$ and $D_{\text{target},i}$.

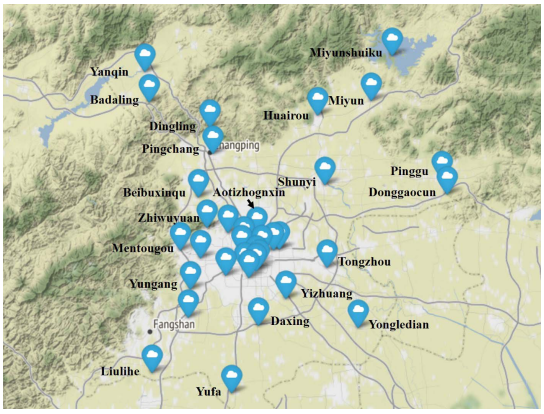


Fig. 4. Distribution of air quality monitoring stations in Beijing.

C. Interpolation Process

The third part of our framework is about the interpolation process. IMA is a missing type-aware framework that adaptively chooses the interpolation method according to the missing data type.

Algorithm 4 shows the complete scheme of IMA. During the data preprocessing process in Algorithm 1, IMA forms neighbor information tables (DT and PNT). DT is used to record the distance between each pair of stations. PNT is applied to obtain the k nearest neighbors according to Pearson's correlation coefficient values. Then, IMA trains multi-XGBoost and LR models for each station. After training, each station s_i starts interpolation operation. Take $PM_{2.5}$ for an example. Assume that s_i samples a measurement $M = [m_1, \ldots, m_q]$ at timestamp t and loss the value of m_j, i.e., $PM_{2.5}$. Before interpolation, s_i needs to obtain spatiotemporal information first. Therefore, s_i inquires the k nearest neighbors in PNT and forms interpolation matrix $I = [d_{ij}]_{w \times (k+1)}$ using neighbors' data in the current time window, where the time interval ranges from $t - \lfloor w/2 \rfloor$ to $t + \lfloor w/2 \rfloor$. What needs illustration is that the local data are stored in the first column in I, the missing data are $d_{\lfloor w/2 \rfloor 1}$. After getting M and I, s_i judges the missing data type according to Definitions 1–3. IMA determines whether the missing type is a spatiotemporal loss, attributes loss, and matrix loss in turn. IMA uses the improved multiviewer method to estimate missing data if the missing type is not a spatiotemporal loss. If the missing type is a spatiotemporal loss but not an attribute loss, the improved multi-XGBoost is used to interpolate. Furthermore, if the missing type contains both the above two types, MF is preferred if there is no matrix loss. In addition, the left missing values are replaced by past values.

V. Experiment

To evaluate the proposed IMA framework, we have conducted several experiments on two real-world datasets. All the algorithms were implemented using Python scripts on a PC with a 2.6 GHz Intel Core i7-1075H CPU, 16-G memory, and the Windows10 operating system.

Algorithm 4 Complete Data Interpolation Method (IMA)

Input:
Original data: D; Size of time window: w
Neighbors size: k
Output:
Final Dataset D

```
Get DT, PNT refer to Algorithm 1
Train XGBoost model, refer to Algorithm 2
Train Linear Regression model, refer to Algorithm 3
for all missing data timestamp t in each station s_i do
  Get current measurement M = [m_1, ..., m_q] and interpolation matrix I = [d_ij]_{w x k+1}.
  Judge missing data type according to Definition 1-3.
  if Not spatiotemporal loss then
    d̂_idw, ψ_idw ← using IDW method, refer to (14), (15)
    d̂_ses, ψ_ses ← using SES method, refer to (16), (17)
    d̂_ucf, ψ_ucf ← using UCF method, refer to (18), (19)
    d̂_icf, ψ_icf ← using ICF method, refer to (20), (21)
    if d̂_idw, d̂_ses, d̂_ucf, d̂_icf all have value then
      d̂ ← get result using LR[i] model, refer to (22)
    else
      d̂ ← get ψ weighted result, refer to (23).
    end if
  else if Not attributes loss then
    Do estimation using multi-XGBoost method
  else if Not matrix loss then
    Do estimation using MF method for interpolation method I
  else
    Replace missing data using past data.
  end if
end for
```

A. Datasets and Ground Truth

The air pollution data during two periods [35], [36] in Beijing are used to verify the performance of IMA, which are from 2017/01/01 to 2018/01/31 (201701_201801) and from 2018/01/31 to 2018/03/31 (201802_201803).

The above-mentioned two air quality data were collected every 1 h from 35 stations in Beijing city, as shown in Fig. 4. Each air quality instance consists of the concentration of six air pollutants, viz., $PM_{2.5}$, PM_{10}, NO_2, CO, O_3, and SO_2. There are over 310 000 (9482×35) in 201701_201801, and over 40 000 (1416×34) in 201802_201803. In addition, the timestamp and position information are recorded in each measurement, which can be used to analyze the spatiotemporal correlation of air pollution data.

Table V shows the percentage of missing data of each attribute. The worst case scenario is that in 201701–201801, about 32.9% of PM_{10} data are missing. Meanwhile, data missing problems of other attributes also cannot be neglected.

To evaluate the performance of our scheme, we partition the original data into two parts: training set and testing set. Since there is no ground truth for those missing values in original data and the patterns of data missing are not random, we applied a mechanism in [26] to form testing sets. First, we get missing data in each month (i.e., 2017/05/01 18:00:00) and then check the next month data (i.e., 2017/06/01 18:00:00) at the corresponding timestamp. Those data in the next month are used as ground truths if they are not NaNs. After forming the testing set, the rest of measurements are used as training set.

TABLE V
STATISTICS ON MISSING VALUES IN EXPERIMENTAL DATASETS

Date	number of records	$PM_{2.5}$	PM_{10}	NO_2	CO	O_3	SO_2
201701-201801	311010	14.2%	32.9%	13.7%	21.0%	14.2%	13.7%
201802-201803	48144	3.7%	24.3%	3.7%	4.3%	4.2%,	3.8%

B. Baselines and Metrics

1) Global Mean (GMean) and Local Mean (LMean) [37]: GMean and LMean are two common strategies for interpolating the missing data by mean of all entries or by mean of the data in the current time window. The time window size is critical to the performance of LMean. In the experiments, we set the time window size of the LMean method as 13 after many attempts.

2) XGBoost [22]: XGBoost is a boosting tree model, which improves its regression accuracy by continuously generating new tree to fit the prediction error. In the experiments, we predict the missing values using the data of other attributes at the same timestamp. In addition, XGBoost uses the same parameters with IMA, which are obtained by the Bayesian optimization algorithm.

3) LightGBM [38]: LightGBM is another boosting tree model, which uses a histogram algorithm to reduce the computation complexity in finding optimal subtree structure and applies a leaf-wise growth strategy to improve the accuracy.

4) K-Nearest-Neighbor (KNN): This method uses an average value of k nearest readings as a prediction. In the experiments, we set the neighbor size as 5, which is the optimal number after conducting many trials.

5) ST-MVL [26]: ST-MVL is a multiviewer learning-based method, which estimates the values using the spatiotemporal correlation of data. During each interpolation, ST-MVL applies four methods to estimate missing values, respectively, and then weights the four estimations to generate a final result. For those block missing data, ST-MVL only uses IDW or SES methods to do the estimation.

6) Accuracy Metrics: We evaluate our framework by mean absolute error (MAE), mean relative error (MRE), and accuracy (ACC)

$$\text{MAE} = \frac{\sum_i |x_i - \hat{x}_i|}{n} \tag{24}$$

$$\text{MRE} = \frac{\sum_i |x_i - \hat{x}_i|}{\sum_i x_i} \tag{25}$$

$$\text{ACC} = 1 - \frac{\sum_i |x_i - \hat{x}_i|}{\sum_i x_i} \tag{26}$$

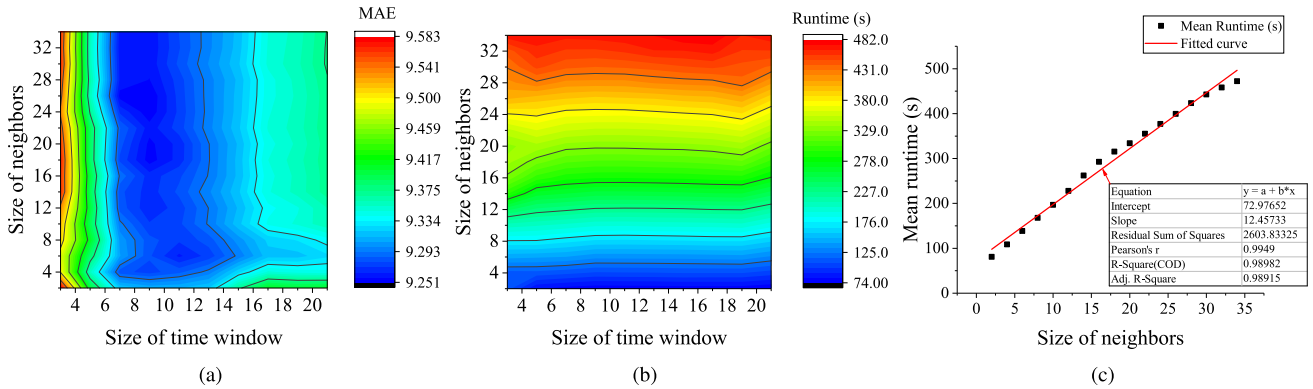


Fig. 5. Results of grid search for proper neighbor size (k) and time window size (w) using data from one station. (a) MAE. (b) Runtime. (c) Mean runtime.

TABLE VI
COMPARISON AMONG DIFFERENT METHODS FOR MAE

Dataset	Attributes	GMean	LMean	XGBoost	LightGBM	KNN	ST_MVL	IMA
201701-201801	PM2.5	35.056	24.775	15.274	15.311	20.271	40.296	**10.702**
	PM10	52.422	34.612	23.213	23.070	24.943	30.926	**20.300**
	NO2	25.028	19.150	13.119	13.125	14.516	19.138	**9.537**
	CO	0.528	0.427	0.259	0.261	0.288	0.727	**0.224**
	O3	50.869	44.063	38.228	38.059	40.844	80.501	**28.254**
	SO2	6.151	4.719	4.003	3.968	4.620	7.164	**2.677**
201802-201803	PM2.5	51.602	19.362	20.900	21.429	27.486	17.118	**10.649**
	PM10	88.551	44.220	58.583	57.193	59.003	35.858	**30.831**
	NO2	24.290	11.995	13.754	13.714	14.837	7.617	**6.569**
	CO	0.481	0.268	0.236	0.232	0.237	0.170	**0.147**
	O3	30.357	17.432	18.221	17.933	21.298	8.619	**7.981**
	SO2	6.136	3.137	4.126	4.139	5.073	2.023	**1.787**

TABLE VII
COMPARISON AMONG DIFFERENT METHODS FOR MRE

Dataset	Attributes	GMean	LMean	XGBoost	LightGBM	KNN	ST_MVL	IMA
201701-201801	PM2.5	0.735	0.520	0.320	0.321	0.425	0.845	**0.224**
	PM10	0.626	0.413	0.277	0.275	0.298	0.369	**0.242**
	NO2	0.618	0.473	0.324	0.324	0.359	0.473	**0.236**
	CO	0.616	0.498	0.302	0.304	0.336	0.847	**0.262**
	O3	0.650	0.563	0.489	0.486	0.522	1.029	**0.361**
	SO2	0.989	0.759	0.644	0.638	0.743	1.152	**0.431**
201802-201803	PM2.5	0.567	0.213	0.230	0.236	0.302	0.188	**0.117**
	PM10	0.619	0.309	0.409	0.400	0.412	0.251	**0.215**
	NO2	0.447	0.221	0.253	0.252	0.273	0.140	**0.121**
	CO	0.434	0.241	0.213	0.210	0.214	0.153	**0.132**
	O3	0.617	0.355	0.371	0.365	0.433	0.175	**0.162**
	SO2	0.554	0.283	0.373	0.374	0.458	0.183	**0.161**

where $\hat{x}_i$ is an estimation value, x_i is the ground truth, and n is the number of samples.

C. Performance Under Different Neighbor Sizes and Time Windows

According to (14)–(21), interpolation results are influenced by time window w and neighbor size k of interpolating matrix. Taking data from air quality monitoring stations for example, the data sampled at near time are closer to the missing data in one station. From this perspective, the value of w should be set as small as possible. However, continuous data missing often occurs, and further, the temporal-based interpolation method will fail due to little or even no temporal information in the interpolation matrix. As for spatial correlation, stations with similar deployment environments tend to collect similar data. Thus, IMA calculates Pearson's correlation coefficient between each pair of stations' data and chooses the top k relevant neighbors for the target station. If possible, the value of k is also the smaller the better because it can avoid the influence of irrelevant data and decrease the computational overhead. However, there are cases where stations lose data at the same time, so small value of k increases the risk of failure of the spatial-based method.

Due to the lack of prior knowledge, we applied the grid search method to find the proper parameters in this article; 201701–201801 is used in this experiment, among which the data of *Aotizhongxin* station are used to construct labeled data using the method described in Section V-A, and the rest of data are auxiliary data. Specifically, the volume of test data is 828,

$w \in [3, 5, \ldots, 21]$, and $k \in [2, 4, \ldots, 34]$. The grid-search results are shown in Fig. 5. From the results, we can draw the following conclusions.

First, MAE values change significantly with w. A smaller time window contains fewer data, resulting in the lack of adequate information for spatiotemporal methods. Here, IMA applies the MF for estimating losing data if elements of the interpolation matrix are not all NaNs. Otherwise, the missing data are replaced by the data of the previous week. Therefore, too small windows have worse MAE. On the other hand, for the nature of air pollution data, data change dynamically in one day. A large size of the time window also decreases the interpolation performance. From the results, the proper w value is between 10 and 12.

Second, because of the spatial correlation between each pair of stations, when the time window size is confirmed, MAE results are affected slightly by the number of neighbors. However, due to local environmental differences, IMA calculates Pearson's correlation coefficient between each pair of stations and selects the most relevant k nearest neighbors as the reference stations, improving the accuracy of estimation. From the results, when k is between 6 and 8, the experimental performances are relatively stable.

Third, Fig. 5(b) shows the runtime of each pair of w and k and Fig. 5(c) shows the relationship between the mean runtime and the number of neighbors. From the results, the running time increases approximately linearly with the number of neighbors.

Parameter Settings: After many tries, we set the value of w as 11 and k as 6 in the following experiments. Furthermore, we set $\alpha = 2$ for IDW, $\beta = 0.8$ for SES, $w = 11$ for SES and ICF, and $k = 6$ for IDW and UCF. In addition, the parameters of multi-XGBoost were set after Bayesian optimization, where the size of estimators, learning rate, and maximum depth was, respectively, set to 210, 0.072, and 9. The training cycle of IMA is set as one year and two months, respectively, for two datasets. We can update the model using the latest data every year or half a year as routine maintenance in practice.

D. Comparative Experiment With Baseline Methods

In this section, we demonstrated the performance of IMA in two datasets by different attributes. We first list the performances for each attribute in Tables VI and VII. Then, the overall scores are shown in Figs. 6 and 7, which are obtained by averaging the results from two datasets for each method.

Due to the narrow concentration range, the MAE performance of CO and SO_2 is lower than other attributes. From the results in Tables VI and VII, though GMean is the easiest method to implement, its performances are almost the worse because the concentrations of atmospheric pollutants such as $PM_{2.5}$ vary dramatically. Since the data from the environmental monitoring systems contain temporal correlation, LMean has lower MAE and MRE than GMean, but it fails to estimate block missing data, whose data size is larger than the time window. For those data, LMean fills the global mean value instead. Therefore, the occurrence of massive data loss will result in

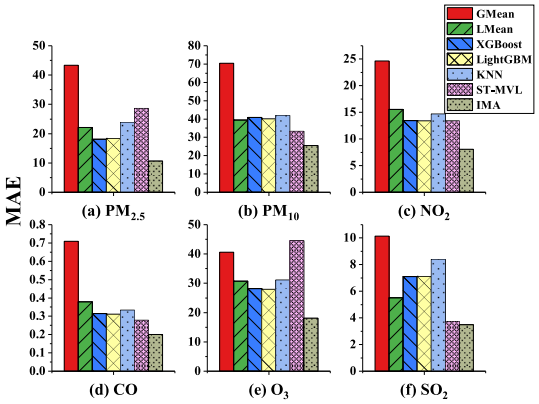


Fig. 6. Overall MAE performance of all methods.

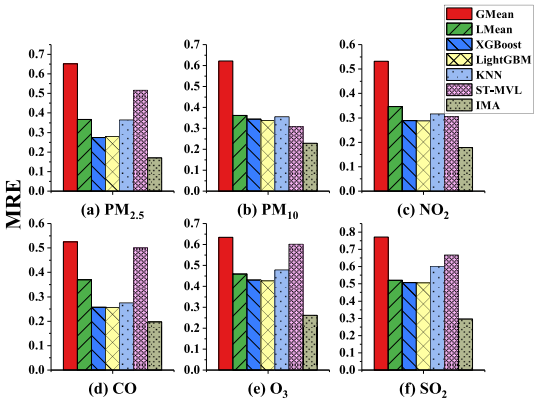


Fig. 7. Overall MRE performance of all methods.

similar performances for GMean and LMean. Two decision-tree-based methods apply the ensemble learning mechanism to mine the correlation between different attributes and form a complex regression model. From the results, both XGBoost and lightGBM have a stable performance. KNN finds the k nearest neighbors in the feature space and estimates the missing value by averaging neighbors' data. From the results, two decision-tree-based methods have a better performance than KNN because they learn the deep correlation between different attributes. ST-MVL is the most related scheme to IMA, which applies four methods to estimate missing data. For the block missing data, ST-MVL uses IDW or SES method to do interpolation. In most cases, such as interpolating data in 201802–201803, ST-MVL has a similar performance to IMA. While comparing the results of interpolating $PM_{2.5}$ in two datasets, the performance of ST-MVL decreases obviously with the increase of the size of the block missing data. Unlike ST-MVL, IMA combines three methods and adaptively chooses different interpolation methods according to the missing data type. Furthermore, if the multiviewer method fails, IMA uses a confidence value to weight the valid results. The results in Tables VI and VII show that IMA significantly improves the performance of ST-MVL in estimating block missing data and outperforms other methods.

Figs. 6 and 7 show the average performances of all methods in two datasets. GMean still has the worst performance in

most cases. Due to the temporal correlation of air quality data, LMean has a similar performance with two tree-based methods. The performance of ST-MVL is sensitive to missing data types. As shown in Fig. 8, we compare the original measured dataset with two operated datasets from six stations, and the two operated datasets are interpolated by IMA and ST-MVL. The data range from 2017-04-01 to 2018-08-01 in Fig. 8(a) and range from 2017-05-16 to 2017-06-04 in Fig. 8(b). Massive data are missing from 2017-05-18 to 2017-03-31. IDW, UCF, and ICF (mentioned in Section II-B) in ST-MVL fail to estimate. Therefore, the use of SES makes the estimated results very smooth. For the losing data, IMA checks whether the data are attributes loss and matrix loss. If so, IMA replaces the missing data using past data. Otherwise, IMA applies the proposed multi-XGBoost or MF to make interpolation.

Furthermore, due to the lack of ground truth in massive data loss, we artificially remove seven-days' $PM_{2.5}$ data from 15 stations in different intervals and make the interpolation using IMA and ST-MVL. The 15 missing areas are from different intervals; in other words, the neighbors' data can be used to interpolate. In this case, ST-MVL uses IDW and SES to make an estimation. From the results in Fig. 9, ST-MVL has a close performance with IMA in Fig. 9(a) and (i). However, in other intervals, ST-MVL has unsatisfactory results. The reasons are as follows. First, ST-MVL uses data from all stations to estimate, while air pollution data contain regional differences; Second, ST-MVL only considers the spatiotemporal correlation of one attribute, which will be infeasible in massive data loss in Fig. 8(b), while IMA improves the robustness of ST-MVL by adaptively choosing an interpolation method according to the missing data type. Consequently, IMA has stable performances in estimating the long-term missing measurements with various change patterns.

The datasets used in this article are sampled 1 h once to satisfy the application purpose. Although higher sampling rates may reduce some missing data while lower sampling rates may increase a little bit of the missing rates, most missing data are caused by a lot of other reasons, such as outliers and noise. Therefore, instead of considering sampling rates, this article directly considers missing rates of data, as shown in Table V. The higher the missing rate, the smaller size of valid data that can be used to predict missing data. From the results in Tables VI and VII, the most related method (ST-MVL) works well to predict missing data on the smaller dataset (201802–201803) with a lower data missing rate, but its performance declines significantly for the bigger dataset (201701–201801) with higher missing rates of data. In contrast, the performance of our proposed IMA is always good and steady for both datasets, that is, it is not impacted much by different missing rates of data.

Complexity Analysis: First, we have defined the symbols used in complexity analysis in Table I.

The main computation overhead of IMA is spent on the model preparation and interpolation processes. There are three tasks during the model preparation process, among which a complexity of $O(n(wk + w + k))$ is required to get LR training dataset, and training LR and multi-XGBoost models requires computational complexity of $O(nq^2)$ and $O(qn \log n + kdqn)$, respectively. IMA analyzes the missing data type during the interpolation process and chooses the corresponding interpolation method: multiviewer method, multi-XGBoost method, or matrix factorization method. The analyzing process requires a complexity of $O(w + q + k)$. For estimating one missing data, the complexity of the multiviewer method, multi-XGBoost method, and matrix factorization method is, respectively, $O(wk+w+k)$, $O(kd)$, and $O(mwk)$. Overall, IMA has a complexity of $O(n \log n)$.

GMean replaces the missing data with the global mean, which has a complexity of $O(n)$ in both training and interpolation processes, while LMean needs no training process. It calculates the mean of data in the current time window and interpolates the missing data using the local mean, which owns a complexity of $O(wn)$ in the interpolation process. XGBoost and LightGBM are two decision-tree-based methods. Let k be the tree number and d be the tree depths. The main computation overhead of XGBoost is spent on sorting the original data, which has a complexity of $O(n \log n)$. Besides, XGBoost requires a complexity of $O(kdqn)$ to construct its decision trees. Unlike XGBoost, LightGBM forms histograms for original data and finds the best split for subtree using the histogram value. Although the prediction accuracy of LightGBM is affected, its computational complexity in forming decision trees is reduced to $O(n + kdqb)$, where b is the bin number of histograms. Both decision-tree models require a complexity of $O(nkd)$ in the interpolation process.

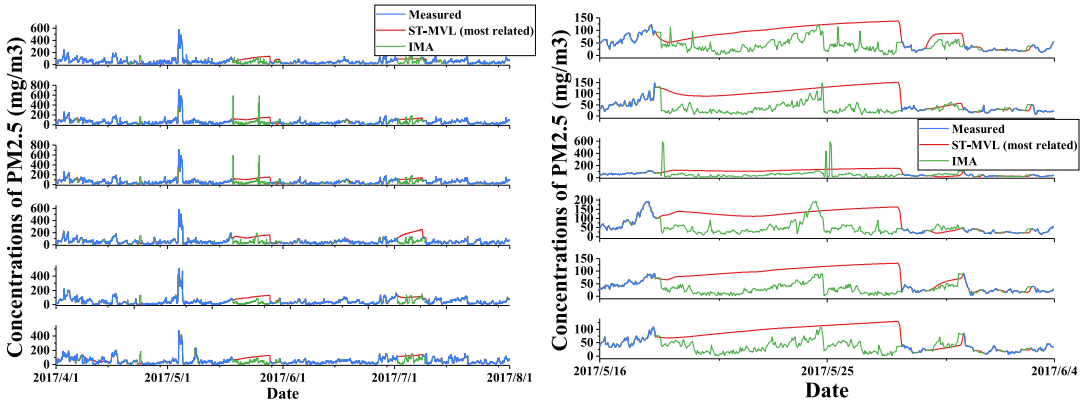
KNN is a distance-based method, which estimates the missing data using the nearest k measurements. During the model preparation process, KNN sorts the original data and applies a KD-tree structure to optimize search time. It totally has a complexity of $O(n \log n + q \log n)$. Therefore, KNN requires a complexity of $O(k \log n)$ for searching k nearest neighbors in the KD-tree. ST-MVL is the most relevant method for IMA, which requires a complex of $O(n(wk + w + k) + nq^2)$ during model preparation and has a complexity of $O(n(wk + w + k))$ during the interpolation process. The complexity of all methods is listed in Table VIII.

TABLE VIII
TIME COMPLEXITY OF METHODS

Method	Training process	Interpolation process
GMean	$O(n)$	$O(n)$
LMean	None	$O(n)$
XGBoost	$O(nlogn)$	$O(n))$
LightGBM	$O(n)$	$O(n)$
KNN	$O(nlogn)$	$O(nlogn)$
ST-MVL	$O(n)$	$O(n)$
IMA	$O(nlogn)$	$O(n)$

TABLE IX
COMPARISON OF TWO INTERPOLATION METHODS ON DEEPAIR FOR MRE

Interpolation method	1h	2h	3h	4h	5h	6h	1-6h
IMA	13.48	18.10	21.72	25.03	28.16	30.87	22.89
Normal	14.04	19.31	23.14	26.75	29.63	32.34	24.20



(a) (b)

Fig. 8. Data before and after interpolation by two methods. (a) Data range from 2017-04-01 to 2018-08-01. (b) Data range from 2017-05-16 to 2017-06-04.

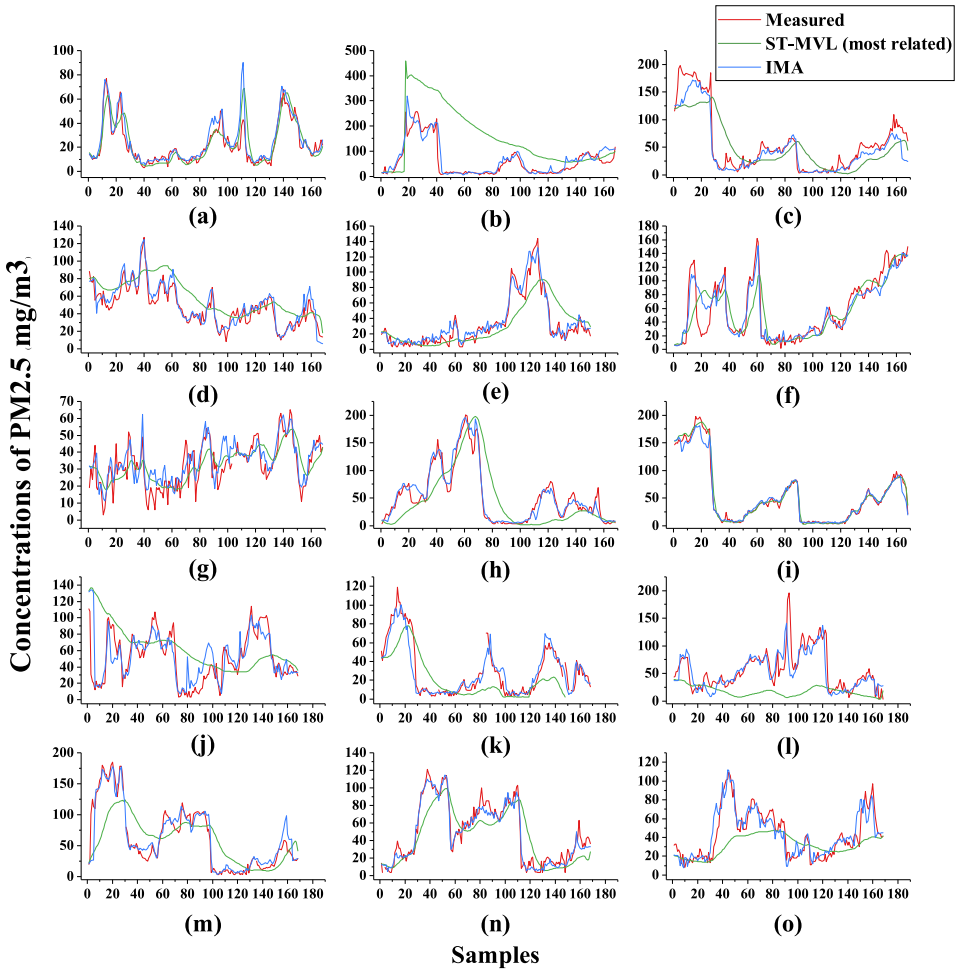


Fig. 9. Interpolation performance for continuous data missing (seven days). (a)–(o) represent 15 different stations, from which we artificially remove seven days of continuous data.

IMA has a higher complexity than other methods except for XGBoost and KNN during the training process. However, after finishing the training, IMA is efficient when the trained model in use. In practice, IMA is designed to train the model on the server with abundant computational resources. We speeded up the model training by the multicore parallel computing technique due to the independence of each model. From the experimental results, the overall performance of our framework outperforms other methods. Therefore, the slow training process will not impact the general performance of our framework.

E. Performance of Different Imputing Methods for Spatiotemporal Model

To further verify that IMA can improve the data quality for prediction tasks, we apply a prediction model in [23],

TABLE X
COMPARISON OF TWO INTERPOLATION METHODS ON DEEPAIR FOR ACC

Interpolation method	1h	2h	3h	4h	5h	6h	1-6h
IMA	0.881	0.841	0.810	0.781	0.754	0.731	0.800
Normal	0.874	0.829	0.796	0.764	0.740	0.716	0.786

called DeepAir. DeepAir is a deep distributed fusion network. It can predict the future concentrations of particular pollutants using pollution data, meteorology data, weather forecast data, and so on. To mine the different influences from various factors, DeepAir constructs five fusion nets and merges all outputs using a parametric-matrix-based fusion method. See more details in [23].

As a comparison method, we added a common interpolation method called Normal, which applies SES on that short-interval loss and replaces long-interval loss with past values. In the data preprocessing stage, two interpolated datasets were formed using the IMA and Normal methods, respectively. In these experiments, all data are from 2017-01 to 2018-03, and interpolated data were split into a training set (90%) and test set (10%) in chronological order. During the training process, those data from the past 6 h were used to predict data from the future 6 h. We used two datasets to train the model and obtained the test results in Tables IX and X. The results show that IMA does improve the prediction accuracy of DeepAir and reduce the MRE value at every moment.

VI. CONCLUSION

In this article, a missing type-aware interpolation framework (IMA) was proposed, which adaptively chose different methods according to the missing data type. The improved multiviewer method can mine the spatiotemporal correlation, the proposed multi-XGBoost was used to learn the correlation between various attributes, and MF for interpolation matrix is a type-insensitive method. For further improving the accuracy of the multiviewer method, IMA applies Pearson's correlation coefficient to find k nearest neighbors for forming the interpolation matrix, not the spatial distance. In addition, we adjust each approach in the multiviewer method to various data losing cases and added its corresponding confidence value. When the multiviewer method fails to estimate, we weight all valid outputs with their confidence value as the final result. Our proposed method (IMA) has a complexity of $O(n \log n)$ and $O(n)$ in the model preparation and interpolation process, respectively, which are the same as the most related counterpart method, ST-MVL, but extensive experiments on two real-world datasets demonstrated that IMA performed better than other methods in terms of accuracy. In addition, we use two interpolation methods to complete the two datasets, and the two completed data from 2017-01 to 2018-03 were merged and used for training the DeepAir model. Compared with ST-MVL, IMA improves the interpolation accuracy from 0.818 to 0.849 in a small dataset and from 0.214 to 0.759 in a large one.

REFERENCES

[1] H. Y. Teh, A. W. Kempa-Liehr, and K. I.-K. Wang, "Sensor data quality: A systematic review," *J. Big Data*, vol. 7, no. 1, p. 11, Dec. 2020.

[2] M. Babazadeh, "Edge analytics for anomaly detection in water networks by an Arduino101-LoRa based WSN," *ISA Trans.*, vol. 92, pp. 273–285, Sep. 2019.

[3] I. G. A. Poornima and B. Paramasivan, "Anomaly detection in wireless sensor network using machine learning algorithm," *Comput. Commun.*, vol. 151, pp. 331–337, Feb. 2020.

[4] X. Wang, L. T. Yang, Y. Wang, L. Ren, and M. J. Deen, "ADTT: A highly efficient distributed tensor-train decomposition method for IIoT big data," *IEEE Trans. Ind. Informat.*, vol. 17, no. 3, pp. 1573–1582, Mar. 2021.

[5] X. Jiang and Z. Ge, "Augmented multidimensional convolutional neural network for industrial soft sensing," *IEEE Trans. Instrum. Meas.*, vol. 70, pp. 1–10, 2021.

[6] Z. Yin *et al.*, "Spatiotemporal fusion of Land Surface Temperature based on a convolutional neural network," *IEEE Trans. Geosci. Remote Sens.*, vol. 59, no. 2, pp. 1808–1822, Feb. 2021.

[7] I. Spinelli, S. Scardapane, and A. Uncini, "Missing data imputation with adversarially-trained graph convolutional networks," *Neural Netw.*, vol. 129, pp. 249–260, Sep. 2020.

[8] J. Holloway, K. J. Helmstedt, K. Mengersen, and M. Schmidt, "A decision tree approach for spatially interpolating missing land cover data and classifying satellite images," *Remote Sens.*, vol. 11, no. 15, p. 1796, Jul. 2019.

[9] L. Qu, J. Hu, L. Li, and Y. Zhang, "PPCA-based missing data imputation for traffic flow volume: A systematical approach," *IEEE Trans. Intell. Transp. Syst.*, vol. 10, no. 3, pp. 512–522, Sep. 2009.

[10] J. Haworth and T. Cheng, "Non-parametric regression for space–time forecasting under missing data," *Computers, Environ. Urban Syst.*, vol. 36, no. 6, pp. 538–550, 2012.

[11] M. K. Gill, T. Asefa, Y. Kaheil, and M. McKee, "Effect of missing data on performance of learning algorithms for hydrologic predictions: Implications to an imputation technique," *Water Resour. Res.*, vol. 43, no. 7, Jul. 2007, doi: 10.1029/2006WR005298.

[12] B. Fallah, K. T. W. Ng, H. L. Vu, and F. Torabi, "Application of a multi-stage neural network approach for time-series landfill gas modeling with missing data imputation," *Waste Manage.*, vol. 116, pp. 66–78, Oct. 2020.

[13] Y. Wang, J. Wang, and H. Li, "An interpolation approach for missing context data based on the time-space relationship and association rule mining," in *Proc. 3rd Int. Conf. Multimedia Inf. Netw. Secur.*, Nov. 2011, pp. 623–627.

[14] G. D'Aniello, M. Gaeta, and T.-P. Hong, "Effective quality-aware sensor data management," *IEEE Trans. Emerg. Topics Comput. Intell.*, vol. 2, no. 1, pp. 65–77, Feb. 2018.

[15] J. Tang, G. Zhang, Y. Wang, H. Wang, and F. Liu, "A hybrid approach to integrate fuzzy C-means based imputation method with genetic algorithm for missing traffic volume data estimation," *Transp. Res. C, Emerg. Technol.*, vol. 51, pp. 29–40, Feb. 2015.

[16] M. Amiri and R. Jensen, "Missing data imputation using fuzzy-rough methods," *Neurocomputing*, vol. 205, pp. 152–164, Sep. 2016.

[17] Y. Li and L. E. Parker, "Nearest neighbor imputation using spatial–temporal correlations in wireless sensor networks," *Inf. Fusion*, vol. 15, pp. 64–79, Jan. 2014.

[18] X. Miao, Y. Gao, G. Chen, B. Zheng, and H. Cui, "Processing incomplete K nearest neighbor search," *IEEE Trans. Fuzzy Syst.*, vol. 24, no. 6, pp. 1349–1363, Dec. 2016.

[19] P. Xu, W. Ruan, Q. Z. Sheng, T. Gu, and L. Yao, "Interpolating the missing values for multi-dimensional spatial-temporal sensor data: A tensor SVD approach," in *Proc. 14th EAI Int. Conf. Mobile Ubiquitous Systems: Comput., Netw. Services*, 2018, pp. 442–451.

[20] X. Song, Y. Guo, N. Li, and S. Yang, "A novel approach based on matrix factorization for recovering missing time series sensor data," *IEEE Sensors J.*, vol. 20, no. 22, pp. 13491–13500, Nov. 2020.

[21] X. Hu, H. Zhang, D. Ma, and R. Wang, "A tnGAN-based leak detection method for pipeline network considering incomplete sensor data," *IEEE Trans. Instrum. Meas.*, vol. 70, pp. 1–10, 2021.

[22] T. Chen and C. Guestrin, "XGBoost: A scalable tree boosting system," in *Proc. 22nd ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2016, pp. 785–794.

[23] X. Yi, J. Zhang, Z. Wang, T. Li, and Y. Zheng, "Deep distributed fusion network for air quality prediction," in *Proc. 24th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Jul. 2018, pp. 965–973.

[24] D. Kondrashov and M. Ghil, "Spatio-temporal filling of missing points in geophysical data sets," *Nonlinear Processes Geophysics*, vol. 13, no. 2, pp. 151–159, May 2006.
[25] L. Pan and J. Li, "K-nearest neighbor based missing data estimation algorithm in wireless sensor networks," *Wireless Sensor Netw.*, vol. 2, no. 2, pp. 115–122, 2010.
[26] X. Yi, Y. Zheng, J. Zhang, and T. Li, "ST-MVL: Filling missing values in geo-sensory time series data," in *Proc. 25th Int. Joint Conf. Artif. Intell.*, 2016, pp. 2704–2710.
[27] C. Garcia, D. Leite, and I. Skrjanc, "Incremental missing-data imputation for evolving fuzzy granular prediction," *IEEE Trans. Fuzzy Syst.*, vol. 28, no. 10, pp. 2348–2362, Oct. 2020.
[28] J. Contreras, R. Espinola, F. J. Nogales, and A. J. Conejo, "ARIMA models to predict next-day electricity prices," *IEEE Trans. Power Syst.*, vol. 18, no. 3, pp. 1014–1020, Aug. 2003.
[29] L. Gruenwald, H. Chok, and M. Aboukhamis, "Using data mining to estimate missing sensor data," in *Proc. 7th IEEE Int. Conf. Data Mining Workshops (ICDMW)*, Oct. 2007, pp. 207–212.
[30] S. Wang, J. Tang, Y. Wang, and H. Liu, "Exploring hierarchical structures for recommender systems," *IEEE Trans. Knowl. Data Eng.*, vol. 30, no. 6, pp. 1022–1035, Jun. 2018.
[31] X. He, J. Tang, X. Du, R. Hong, T. Ren, and T.-S. Chua, "Fast matrix factorization with nonuniform weights on missing data," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 31, no. 8, pp. 2791–2804, Aug. 2020.
[32] B. Fekade, T. Maksymyuk, M. Kyryk, and M. Jo, "Probabilistic recovery of incomplete sensed data in IoT," *IEEE Internet Things J.*, vol. 5, no. 4, pp. 2282–2292, Aug. 2018.
[33] G. Y. Lu and D. W. Wong, "An adaptive inverse-distance weighting spatial interpolation technique," *Comput. Geosci.*, vol. 34, no. 9, pp. 1044–1055, Sep. 2008.
[34] X. Su and T. M. Khoshgoftaar, "A survey of collaborative filtering techniques," *Adv. Artif. Intell.*, vol. 2009, no. 12, pp. 1–19, Oct. 2009.
[35] Y. Zheng *et al.*, "Forecasting fine-grained air quality based on big data," in *Proc. 21th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Aug. 2015, pp. 2267–2276.
[36] S. Al-Janabi, M. Mohammad, and A. Al-Sultan, "A new method for prediction of air pollution based on intelligent computation," *Soft Comput.*, vol. 24, no. 1, pp. 661–680, Jan. 2020.
[37] B. Wang, Z. Yan, J. Lu, G. Zhang, and T. Li, "Deep multi-task learning for air quality prediction," in *Neural Information Processing*, L. Cheng, A. C. S. Leung, and S. Ozawa, Eds. Cham, Switzerland: Springer, 2018, pp. 93–103, doi: 10.1007/978-3-030-04221-9_9.
[38] G. Ke *et al.*, "LightGBM: A highly efficient gradient boosting decision tree," in *Proc. 31st Int. Conf. Neural Inf. Process. Syst. (NIPS)*, Long Beach, CA, USA. Red Hook, NY, USA: Curran Associates, 2017, pp. 3149–3157.

Lingqiang Chen received the bachelor's degree in Internet-of-Things engineering from the Zhejiang University of Technology, Hangzhou, China, in 2017. He is currently pursuing the Ph.D. degree with the School of Artificial Intelligence and Computer Science, Jiangnan University, Wuxi, China.

His current research interest is anomaly detection in wireless sensor networks.

Guanghui Li received the Ph.D. degree in computer science from the Institute of Computing Technology, Chinese Academy of Sciences, Beijing, China, in 2005.

He is currently a Professor with the School of Artificial Intelligence and Computer Science, Jiangnan University, Wuxi, China. He has authored over 70 articles in journals or conferences. His research interests include wireless sensor networks, fault-tolerant computing, and nondestructive testing and evaluation. His research was supported by the National Natural Science Foundation of China, the Jiangsu Provincial Science and Technology Foundation, and other governmental and industrial agencies.

Guangyan Huang (Member, IEEE) received the Ph.D. degree in computer science from Victoria University, Footscray, VIC, Australia, in 2012.

She was an Assistant Professor with the Institute of Software, Chinese Academy of Sciences, Beijing, China, between 2007 and 2009, and visited the Platforms and Devices Center, Microsoft Research Asia, in the last half of 2006. She is an Associate Professor with the School of Information Technology, Deakin University, Melbourne, VIC, Australia. She has authored over 110 publications mainly in data mining, the IoT/sensor networks, text analytics, image/video processing, and spatiotemporal data analytics.

Dr. Huang was a recipient of the ARC Discovery Early Career Researcher Award (DECRA) fellowship and a chief investigator of two ARC Discovery Projects.

Pei Shi received the master's degree in system analysis and integration from the Nanjing University of Information Science and Technology, Nanjing, China, in 2014, and the Ph.D. degree in control science and engineering from Jiangnan University, Wuxi, China, in 2020.

She is a Lecturer with the Binjiang College, Nanjing University of Information Science and Technology, Wuxi. Her research interests include wireless sensor networks, edge computing, and artificial intelligence.