Journal of Universal Computer Science, vol. 30, no. 6 (2024), 814-846
submitted: 22/4/2023, accepted: 14/1/2024, appeared: 28/6/2024

IMD-MP: Imputation of Missing Data in IoT Based on Matrix Profile and Spatio-temporal Correlations

G.V.Vidya Lakshmi
(School of Computer Science and Engineering, VIT-AP University, Amaravathi. Andhra Pradesh, India
https://orcid.org/0000-0003-1279-9684, vidyalakshmi.21phd7089@vitap.ac.in)

S. Gopikrishnan
(School of Computer Science and Engineering, VIT-AP University, Amaravathi. Andhra Pradesh, India
https://orcid.org/0000-0001-9082-9012, gopikrishnan.s@vitap.ac.in)

Abstract: Data in the Internet of Things (IoT) domain may be missing due to connectivity errors, environmental extremes, sensor malfunctions, and human errors. Despite the many approaches for imputing missing values, the most significant difficulty in terms of imputation precision or compute complexity for larger missing sub-sequences in uni-variate series is still being explored. This work introduced IMD-MP (Imputation of Missing Data using Matrix Profile), a new technique that improves imputation accuracy for big data analysis in IoT applications based on spatial-temporal correlations using a novel distance metric Matrix Profile Distance (MPD). Our method preserves spatial correlation by grouping the sensors present in the network (using grouping algorithm-GA) to impute the missing data of the failed sensor node. After grouping, similar sensor nodes to the failed sensor node are identified using the Node Similarity Algorithm (NSF). From its similar sensor data, a certain number of sub-sequences that are most similar to the one preceding the failed node's missing values are gathered. These sub-sequences heights are optimized to ensure temporal correlation in the imputed data. To find the optimal imputation sequence, the current research uses MPD and similarity scores. Numerical findings using sensor data from real-time environmental monitoring and Intel data sets demonstrate the algorithm's effectiveness compared to other benchmarks.

Keywords: Internet of Things, Data imputation, Univariate data, Spatial correlation, Temporal correlation, Data quality
Categories: H.3.1, H.3.2, H.3.3, H.3.7, H.5.1
DOI: 10.3897/jucs.105363

1 Introduction

In recent times, the Internet of Things (IoT) has experienced significant advancements in key technologies. As a result, IoT-based platforms and systems have been extensively utilised in diverse disciplines and industries. These include intelligent transportation [Djenouri et al. 2022], smart buildings, the healthcare sector [Mohammed et al. 2023], positioning and navigation [Elsanhoury et al. 2022], smart grids [Syu et al. 2023], and the logistics industry [Turabieh et al. 2019]. IoT applications commonly consist of multiple nodes, each of which has various sensors. These nodes are distributed throughout vast regions to monitor specific physical processes or environment. The sensors at these nodes enable the collection of vast amounts of data within the physical environment, forming the foundational basis for various IoT applications.

The collected datasets play a vital role in extracting meaningful knowledge for various purposes, including classification, pattern recognition, trend analysis, and decision-making [Amin et al. 2022]. However, dealing with datasets containing missing values presents a challenge in generating the required information for intelligent decision-making [Ni and Cao 2022]. The efficacy of IoT systems in making accurate decisions is based upon the quality of the data collected. In order to meet the requirements of many applications, it is necessary to have datasets without any missing values. This is important because missing values can negatively impact the performance of algorithms and make them unusable [Da Silva et al. 2020].

1.1 Impacts of Missing data with large gaps in IoT

Data Integrity Consider a smart home IoT system that collects temperature readings every hour. If significant gaps exist in the temperature data due to sensor malfunctions or network connectivity issues, accurately determining temperature patterns or detecting anomalies in the home environment becomes challenging [Naik et al. 2023].

Analysis and Decision Making In a predictive maintenance system for industrial machinery, the presence of missing data gaps can disrupt the analysis of machine performance trends. Without complete data, accurately predicting when a machine requires maintenance or making informed decisions regarding maintenance schedules becomes challenging [Jeong et al. 2023].

Prediction and Forecasting Consider a weather monitoring IoT network that measures various environmental parameters [Kim et al. 2023]. If there are significant missing data gaps in temperature measurements over an extended period, it can affect weather forecasting models, leading to inaccurate predictions of temperature trends and weather conditions.

System Performance and Optimization In a smart grid system that optimizes energy distribution, missing data gaps in electricity consumption readings can impact load balancing and energy management algorithms [Schreiber et al. 2023]. Without complete data, the system may not accurately allocate energy resources, leading to sub-optimal performance and inefficient energy usage.

Data Imputation Challenges In a healthcare IoT application, suppose a patient monitoring system collects vital signs data at regular intervals. Imputing missing values becomes hard when there are long periods of data gaps caused by sensor failures or connection issues [Zou et al. 2020]. The imputation process must carefully consider the temporal as well as spatial nature of the data to avoid introducing biased or incorrect values.

These examples highlight how missing data with large gaps can affect various aspects of IoT applications, including data integrity, analysis, decision making, prediction, and system performance. Addressing these impacts requires careful consideration of data collection strategies, robust imputation techniques, and the potential consequences of missing data on the overall application.

1.2 Importance of Data Imputation

All the data from IoT devices, like data from sensors, actuators, software logs, and business data, must meet real-time requirements so that the data can be analysed later. Various methods, including computational intelligence and artificial intelligence, can be employed for data analysis. However, these methods require complete data for accurate

quality estimation [Aldoseri et al. 2023]. The presence of missing data poses a hurdle, preventing the generation of intelligent and informed decisions based on the data derived from IoT.

The IoT system must effectively identify and address missing data to ensure accurate resource allocation and decision-making [Villalonga et al. 2020]. The presence of missing data disrupts resource allocation [Dong et al.2023], leading to sub optimal utilization and potential inefficiencies. This increased latency compromises the system's responsiveness to changing conditions and introduces the risk of biased resource allocation [Mohajer et al. 2023], affecting fairness and equity. Furthermore, compromised predictive analytics impede the system's ability to forecast future demands, exacerbating challenges in optimal resource distribution [Mohajer et al.2022].

Often, a commonly used method for dealing with missing values entails either removing or disregarding them and re-sending the missing data until it is received without any issues. Nevertheless, each of these approaches possesses its own drawbacks [Alsufyani et al. 2021], [Majidi et al. 2021]. Alternatively, the process of imputing missing data can be done by replacing them with either the mean value, median value or a constant, which is a widely adopted method. This strategy fails to account for the existing correlations among features and does not adequately examine the data distribution.

Imputation techniques serve as a valuable approach to address missing data, enabling the analysis of incomplete datasets and enhancing the reliability and effectiveness of IoT applications across diverse domains [Song and Szafir 2018]. Missing data reconstruction can help to:

- **Improve decision-making:** Missing data can lead to incorrect decisions, but by reconstructing the missing data [Atiquzzaman et al. 2020], the accuracy of decision-making can be improved.
- **Increase system efficiency:** Missing data can cause delays or errors in system operations, but by reconstructing the missing data, system efficiency can be improved.
- **Enhance system safety:** In safety-critical applications such as transportation or healthcare, missing data can lead to dangerous situations. By reconstructing missing data, system safety can be enhanced.
- **Reduce maintenance costs:** By identifying and addressing missing data, maintenance costs can be reduced by avoiding unnecessary sensor replacements or repairs.

1.3 Motivation

In this study, we provide a novel imputation algorithm based on the Matrix Profile Distance (MPD) [Lee et al. 2021] measure that fills in large gaps in uni-variate time series of IoT imputation. This research is motivated by the critical need to enhance the accuracy of IoT data imputation by harnessing both spatial and temporal correlations. Spatial correlations, pertaining to the interdependence between different sensors within a network, are often overlooked in existing methodologies. To achieve this the suggested method utilized the new distance metric called Matrix Profile Distance (MPD). MPD can be quickly calculated, providing for a more time-efficient comparison of two time series data, while the DTW metric is not as resistant to noise with inconsistencies, etc.

The MPD has proven effective in capturing temporal patterns and anomalies. However, current applications of MPD in imputation often lack a comprehensive exploration of spatial correlations across sensors. By concentrating on both spatial and temporal

dimensions concurrently, our research aims to bridge this gap and provide a robust imputation framework for handling missing IoT data. As the proposed method follows uni-variate imputation, it does not consider the correlations among the variables. Whereas in multivariate imputation, the imputation of data is performed based on the correlations among the features that need to be reconstructed.

1.4 Research contributions

For uni-variate IoT data with large missing gaps, we introduced a new method for Imputation of Missing Data using the MPD measure along with spatial and temporal correlations. The following are the novel contributions in paper:

1. We take advantage of the MPD [Gharghabi et al. 2020] principle in IMD-MP (Imputation of Missing Data using MPD) algorithm to identify the most similar pattern(s) from same sensor or neighboring sensors for the imputation purpose.
2. To search for similar patterns in neighbouring nodes this paper suggests a GA (Grouping Algorithm) which groups the set of sensor nodes based on information available at each node. This GA helps to preserve the spatial correlation in imputed data.
3. After grouping, if 'n' sensors were present in a particular group , Node Similarity Finding (NSF) algorithm finds 'm' sensor nodes that are most similar to failed node (node which is having missing data) such that $m < n$. This filtration helps to reduce the computational complexity because searching for similar patterns in the entire group of nodes may have more computational complexity than searching in 'm' most similar sensors.
4. To preserve temporal correlations in imputed data, this research conduct optimal displacement of the imputation sub-sequences in tandem of the y-axis, leading to improved similarity percentage and decreased WMAPE and NMAE.
5. The proposed IMD-MP algorithm significantly performed better than existing imputation techniques after applying on two datasets 1. Intel dataset and 2. Real world dataset with a seasonal component.
6. The findings illustrate that IMD-MP can be effectively calculated and has the ability to outperform any algorithms based on DTW and MPD (that utilised data from a single sensor).

The paper is organized as follows: Section 2 provides an overview of existing works on IoT data imputation. In Section 3, we present a detailed description of the proposed method. The evaluation metrics, baseline methods, and experiment setup are outlined in Section 4. Section 5 delves into a comprehensive analysis of the performance of the proposed method. Finally, conclusions are drawn in Section 6 based on the findings discussed throughout the paper.

2 Related work

This work offered a missing data estimate algorithm based on temporal and geographical correlations in light of the fact that existing algorithms for estimating missing data do

not make full use of sensor data features and have high computational cost and low accuracy. The primary purpose of the imputation method is to impute data with minimal error while keeping the data's structure, pattern, and trend. Here, we briefly describe various cutting-edge strategies for data imputation in IoT with their limitations. The existing methods for missing data imputation were briefly categorised according to the mechanism being used.

2.1 Basic Techniques in Data Imputation

Expectation Maximization Imputation(EMI): Through Expectation (E) and Maximization (M) stages, the imputation parameters in EMI are determined optimally. EMI [Deng et al. 2022] iteratively estimates missing data in the Expectation step and re - evaluates the maximum-likelihood estimate based on the Maximization step. Until EMI converges, this procedure is repeated, or until there are no significant changes between iterations. EMI has slow convergence and also converges to the local optimum only.

MissForest: In the same way that random forest classification uses the bagged ensemble, in MissForest also every tree being trained on a various subset of X_{obs} (observed data) [Turabieh et al. 2019]. The ensemble's finalized outcome will be the aggregate of the individual trees outputs. The problem with MissForest is, it is not able to run on small datasets effectively.

Multivariate Imputation by Chained Equations(MICE): The parameters of MICE have also been estimated using a random forest model. In [Doove et al. 2014] the authors have suggested recursive partitioning as a means to keep track of data interactions when employing MICE. The effectiveness of Variational Autoencoder (VAE), Multiple Imputation by Chain Equations (MICE), Neural Network with Random Weights (NNRW), K-Nearest Neighbor (KNN) and Random Forest-based Imputation (missForest) in achieving effective sensor calibration in the context of missing data was examined in [Okafor and Delaney 2021]. Though MICE is having flexibility, it may not minimise the use of time or resources.

2.2 Deep Autoencoder based Data Imputation

In order to rebuild missing values, [Huamin et al. 2020] suggested a de-noising autoencoder (DAE) based on a time series data representation. The basic working of a deep autoencoder involves two main processes encoding and decoding [Ageng et al. 2021]. For IoT systems with unreliable data sources, [Kök and Özdemir 2020] recommended Deep-MDP. DeepArch, an actual test-bed composed of cloud, fog, and edge layers, was also developed in [Kök and Özdemir 2020]. For the purpose of modelling multidimensional representations with missing entries, [Wu et al. 2022] created the Multi-Attention Tensor Completion Network (MATCN). The study [Du et al. 2019] presented a deep learning technique, the Deep Belief Network (DBN), that was used to recover lost data in sensor networks by considering the spatial-temporal correlation of vast amounts of sensor data. DBN was Computationally intensive because it require a lot of computational power, particularly during the training phase. To train a DAE, you need a large dataset of clean, noise-free data. DAEs are designed to remove a specific type of noise from the input data. If the noise in the data is different from what the DAE was trained on, it may not perform well. DAEs are most commonly used for image and audio processing applications. They may not be as effective for other types of data, such as text or numerical data.

2.3 Data Imputation with Generative Adversarial Imputation Network (GAIN)

The core components of GAIN ([Yoon et al. 2018], which facilitates imputation were Generator and Discriminator. In order to effectively handle relatively independent solar time series data, the source of a Generative Adversarial Network (GAN) was adjusted in SolarGAN, a multivariate solar data imputation method used in [Zhang et al. 2020]. By showing that the optimal GAN imputation was achieved for the Extended Missing At Random (EMAR) and the Extended Always Missing At Random (EAMAR) mechanisms in addition to the naive MCAR, [Deng et al. 2022] proposed Conditional Imputation GAN.

A missing data imputation methodology based on GANs was presented in [Bagchi et al. 2022] adaptive Vector Auto Regressive (VAR) method for missing data imputation in real-time applications, and it made use of recursive least squares. [Turabieh et al. 2019] presented a method for missing value estimation using D-ANFIS (dynamic adaptive network-based fuzzy inference system). In [Chen et al. 2022] proposed a imputation model by combining Low-Rank Tensor Completion (LRTC) with sparse self-representation. Yet, anomalous data values have a significant effect on these imputation procedures made possible by machine learning, deep learning because these require sufficient amount of training data without missing values[Zhang et al. 2020], [Deng et al. 2022], [Bagchi et al. 2022], [Deng et al. 2022], [Turabieh et al. 2019], and in [Chen et al. 2022].

2.4 Spatial and Temporal correlation based Data Imputation

Relying on the MLR (Multiple Linear Regression) model and spatio-temporal correlation, the research [Zaid et al. 2021] presented a technique for recovering missing data to recover lost data at different levels of granularity. In MLR, the complexity increases with increase in number of sensors in the group and also not addressed large missing gap. By reducing the error estimation and increasing the accuracy, an effective spatial data recovery (ESDR) strategy was developed for CPS in [Nower et al. 2013]. For CPS with stochastically incomplete feedback, [Nower et al. 2014] proposed a data recovery approach called ETSDR (Efficient Temporal and Spatial Data Recovery).

In [González-Vidal et al. 2020], presented Bayesian Maximum Entropy (BME), which incorporated the spatio-temporal aspect of IoT data and the volatility of the data received by sensors (BME) to estimate missing values across a range of IoT use cases, gathering data from a mix of low- and high-precision sensors. The BME technique handled data uncertainty very well in its estimation and also required less computational time but this method not addressed imputation for large missing gap for distributed sensor nodes and also requires seasonality in each segment of data in the large missing gap.

In order to perform spatio-temporal missing data imputation, a method STGNN-DAE was created by [Kuppannagari et al. 2021] using the power grid topology and time series data received from the metering infrastructure in the grid as inputs. STGNN-DAE performed imputation by taking into account both temporal and spatial correlations suitable for smart grid application and needed with complete training data as it was using DAE with spatial and temporal correlations. Ratio-Based Imputation (RBI) was developed to fill in large amounts of missing data by integrating data fusion and data mining approaches [Deepak Adhikari et al. 2021]. As RBI was multivariate imputation method it requires high correlations among the features of dataset that is being imputed for missing data.

Data fusion was used for analysis, and data mining was used for imputing. To safeguard intrusion categorization systems from missing scores, [Razavi-Far et al. 2021] looked into the efficacy of missing data imputation algorithms and hybrid intrusion classification system that utilised multiple cutting-edge imputation techniques was developed. The authors in [Jiang et al. 2021] suggested exploiting the topological information in the product graph to impute incomplete data in wireless sensor networks, inspired by the emerging field of Graph Signal Processing (GSP).

Most of the existing methods did not address the imputation of large missing gaps such as [Nower et al. 2014], [González-Vidal et al. 2020] and [Deng et al. 2022]. Some methods that suggested reconstruction for large missing gap still require high sensitive seasonality components. Periodic traffic type feedback imputation was considered which may not suitable for time-critical data patterns and also for data with large missing gap [Nower et al. 2014], [Jiang et al. 2021]. After applying their method to two data sets, the traffic dataset which was not having smoothness lead to large imputation error. The authors of [Hallaji et al. 2021] evaluated their deep neural network-based approach using a variety of reputable imputation techniques, combining the strengths of de-noising auto encoders with those of a ladder architecture. With this technique, a functioning cyber-physical system is analysed with real-world datasets DLIN considered spatial and temporal correlational separately which reduces the imputation accuracy.

The research [Liu et al. 2020] provided an iterative framework using multiple segmented gap iteration, termed Itr-MS-STLecImp, to deliver the best appropriate values when missing data was present in a univariate time series. In this method the for data reconstruction it mainly used the data of sensor whose data was missing. If that sensor failed and not having sufficient data to imputed then this method not able to impute large gaps. The performance of the this method depends on imputation performance of one segment depends on previous segment imputation accuracy. And also requires seasonality in each segment of data in the large missing gap.

Online VTN imputation, based on virtual temporal neighbours, was introduced in [Deng et al. 2022]. First, the computing method and VTN(Virtual Temporal Neighbour) for the stream of sensor data were defined. Following that, the VTN imputation algorithm was shown, which made use of VTN to estimate missing data via regression. The performance of this method depends on number of VTN selected. If there was large missing gap so that large number of VTN selection was not possible. The regression process applied on VTNs requires more computational time also. Authors in [Caillault et al. 2020] suggested a method called Warping-based Imputation using Dynamic Time (DTWBI), which combined the extraction algorithm for shape-feature with the DTW distance metric to fill in massive breaks in uni-variate series. On the other hand, DTWBI has a drawback in that it was very computationally intensive and having less imputation precision. However, one drawback that is shared by all of these techniques is that their reliability suffers when there are a large number of missing gaps in the data. In addition, the imputation accuracy of these methods, which were explained previously, decreased significantly when the data contained a high missing gap. Despite the fact that numerous algorithms for imputing missing values have been developed, the problem pertaining to imputation accuracy or complexity of computations for large missing sub sequences is yet under discussion. The IoT dataset's biggest difficulty is the large number of data points loss in sequence.

2.5 Problem statement

The research gap identified in the existing methods is that only few of them delt with lagre missing gaps. Even they addressed the imputation of large missing gaps, they were complex and having less imputation precision. The existing solutions were very sensitive to noise present in the data. And the state of art methods does not used the combination of spatial and temporal correlations effectively.

The proposed IMD-MP tried to impute the large missing gaps in IoT time series data using MPD. As already MPD based imputation was proposed by [Lee et al. 2021], it lacks spatial correlation in reconstructed data.The MPDist was based on single sensor data for data imputation i.e the observed data of sensor of which the data being missed. The drawback of MPDist imputation was that it may result in inaccurate imputed data if the sensor was not having enough sensed data. To solve this problem, the current research considered both spatial and temporal correlations.

The data sensed by a sensor at node N_i can be a time series denoted by $K = k_1, k_2, k_3 \ldots k_n$. This node N_i is having 'n' other sensors as neighbors. From the time gap between 'p' to 'q' the data observed is $k_p, k_{p+1}, k_{p+2}...., k_q$ is a sub sequence at N_i. If this data ($k_p, k_{p+1}, k_{p+2}....., k_q$) is missed and k_p', k_{p+1}', k_{p+2}'..., k_q' are the estimated sub sequence, then reducing $|k_p - k_p'|$ for all 'p' to 'q' is the missing data imputation problem. Here, k_p', k_{p+1}', k_{p+2}'...., k_q' can be obtained by shifting $k_s, k_{s+1}, k_{s+2}....., k_t$ to optimal height. The sub sequence $k_s, k_{s+1}, k_{s+2}....., k_t$ is selected from number of available sequences from spatially correlated 'm' sensor nodes which 'm' are more similar to N_i and m<n. For selecting 'm' most similar sensors nodes, similarity finding algorithm is required. For picking most similar sequence from 'm' sensor nodes similarity score function is introduced in the later sections. The aim of this research is to reduce the difference between $k_p, k_{p+1}, k_{p+2}....., k_q$ and k_p', k_{p+1}', k_{p+2}'....., k_q' as less as possible with required accuracy.

3 IMD-MP: Imputation of Missing Data using Matrix-Profile Distance

The suggested method consists of three phases: Grouping Phase, Node Similarity Finding (NSF), and Imputation of Missing Data using MPD (IMD-MP). Figure.1 is an illustration of the operational flowchart for the suggested approach. The data-driven IoT system for monitoring requires four primary components to function: Sensor nodes that gather the data, a cloud where the data can be delivered and saved using the communicating protocol, an IMD-MP to reconstruct the missing values of uni-variate data to guarantee persistent decision-making, and a real-time dashboard displaying the amount of missing sequences and the whole reconstructed data with the missing sequences packed in. This system is simple to comprehend and applicable to IoT-based applications in the real world scenarios.

In Algorithm 1, the sensor nodes of the entire network are grouped based on their location information. Consequently, sensor nodes that monitor the same action or have overlapped sensing ranges will be clustered together. The output of Algorithm 1 is then sent to NSF in order to identify the nodes most similar to the failed node in each group. After the NSF phase, the IMD-MP is used to reconstruct the missing values of failed nodes by utilizing similar sensors data.

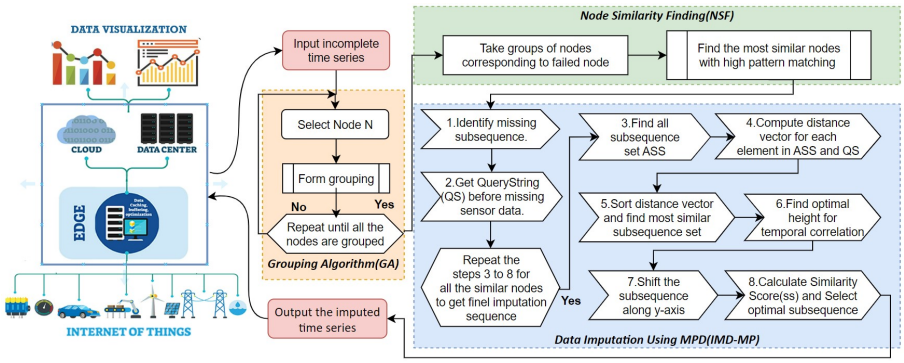


Figure 1: Proposed IMD-MP architecture

3.1 Grouping Algorithm (GA)

This algorithm contributes to maintaining spatial correlation in the imputed data. Spatial correlation in an IoT sensor network refers to the statistical relationship between sensor measurements taken at different locations within the network. In other words, it refers to the degree of similarity or dissimilarity among the sensor data collected from various spatially distributed sensors.

In an IoT sensor network, multiple sensors are deployed across a physical area to monitor different parameters, such as temperature, humidity, pressure, motion, etc. Due to the physical proximity of these sensors and the shared environmental factors, the measurements obtained from nearby sensors often exhibit similar patterns or trends.

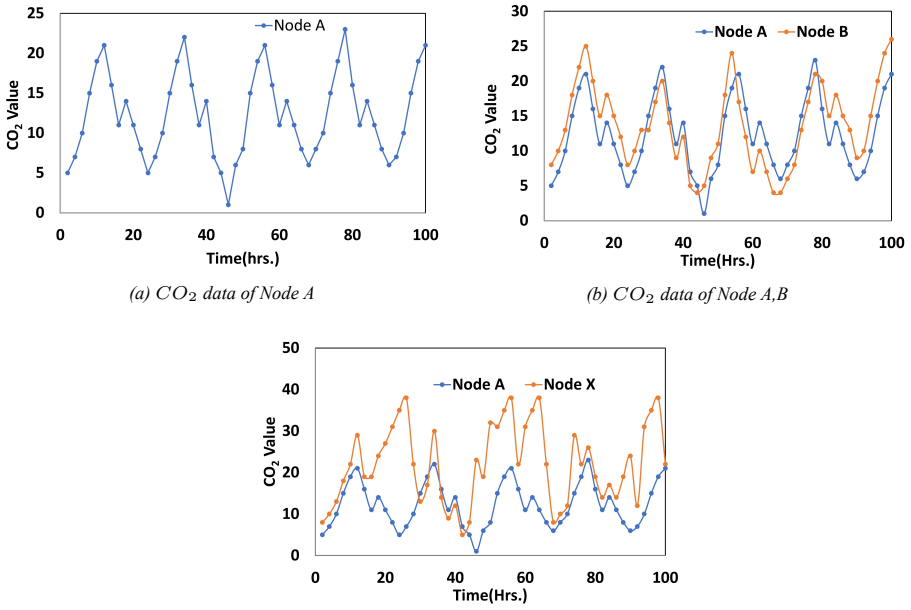
The discrete information at each sensor node in the network, including its position (neighboring nodes, location coordinates, and sensing range), can be utilized to create physical clusters across the entire network by grouping algorithm (e.g., the GP algorithm in the proposed method). The manner of grouping is carried out in response to the mobility of the terminal nodes and the type of application being used. As the purpose of IoT is to monitor physical processes, the basic part of IoT is the WSN, which is placed in the physical system. This process of grouping is carried out only once in static WSNs. Mobile WSNs execute the grouping algorithm periodically to adapt to the changing nature of the network topology.

Figure.3a contained several sensors that were put in close proximity to one another. Twenty sensor nodes are spread out at random throughout the IoT system (for environment monitoring). For example, if the algorithm 1 is applied here, it may pick sensors like 5, 6, 15, and 4 at random from the network to build clusters. It can be observed that sensor node 5 and sensor node 6 share two sensor nodes as neighbors. It is also evident that members of each group are geographically correlated, meaning that their measurements can be used interchangeably to retrieve missing data.

In the Grouping Algorithm 1, the initial node set N is given as input. From that group, a random node n_i is selected as the group head, and its neighboring nodes will group and store the groups in G[][], this process is continued until all the nodes in N are grouped. Here, the common neighboring nodes can be obtained by using the equation 3.1.

$$\mathbf{CNS}\left(\overline{N_i}, \overline{N_j}\right) = \left|NS\left(\overline{N_i}\right) \cap \left|NS\left(\overline{N_j}\right)\right|\right. \tag{3.1}$$

The equation 3.1 finds the common neighbours of two nodes $\overline{N_i}, \overline{N_j}$ by perform-



(a) CO_2 data of Node A

(b) CO_2 data of Node A,B

(c) CO_2 data of Node A,X

Figure 2: Long-term distribution of CO_2 data

ing intersection operation on $NS\left(\overline{N_i}\right), NS\left(\overline{N_j}\right)$ where $NS\left(\overline{N_i}\right), NS\left(\overline{N_j}\right)$ are the neighbouring node sets of nodes $\overline{N_i}, \overline{N_j}$ respectively.

Algorithm 1 Grouping Algorithm(GA)

INPUT: All sensors of the network $N = n_1, n_2....n_n$.
OUTPUT: List of groups G[][]

```
Begin
Initialization: set G ← [ ] [ ]
for i ∈ n do
    get-random n_i
    if n_i in G then return
    else
        G.append(n_i)
    end if
    neighbours=n_i.neighbours
    for (j ∈ neighbours ) do
        common-neighbours=CNS(j,n_i)
        G[n_i].append(common-neighbours)
    end for
end for
End
```

3.1.1 Working of GA

1. Initialization: The algorithm initializes an empty list G to store the groups of sensors.

2. Random Selection: Randomly a sensor n_i in the network will be selected from node set N

3. Group Formation: The algorithm then checks if the randomly selected sensor n_i is already present in any existing groups G. If n_i is already a cluster head, the algorithm returns without making any further modifications. This step ensures that the same sensor is not included as a duplicate, preventing redundancy.

4. Add Sensor to Group: If the sensor n_i is not cluster/group head, it is added as a new group.

5. Identifying Common Neighbors: For each sensor n_i, the algorithm identifies its neighboring sensors ($neighbours$). Then, it iterates over these neighbors (j) and finds their common neighbors with the sensor n_i (using the equation 3.1). These common neighbors represent the spatial correlation between n_i and its neighboring sensors.

6. Adding Common Neighbors to Group: The identified common neighbors are then appended to the group corresponding to sensor n_i. This step ensures that sensors with common spatial characteristics are grouped together, preserving spatial correlation within the group.

After grouping sensors, if there is any missing data in any sensor node in a particular group, then NSF algorithm 2 gives most similar nodes to the node whose data need to be imputed. In the next subsection the working NSF is presented.

3.2 Node Similarity Finding (NSF) algorithm

In the event of a sensor node failure within a network of sensors, optimizing the use of available data becomes imperative. Leveraging data from neighboring sensor nodes and utilizing historical time-series data enables the tracking and restoration of lost data. However, it's essential to note that not all adjacent nodes may exhibit similar data patterns. Figure 2a illustrates the long-term CO_2 distribution from a single sensor node A. Figures 2b and 2c present the simultaneous CO_2 distribution from two nodes, revealing potential differences in the trend of change between them. While Nodes A and B may demonstrate similar patterns of change, Nodes A and X may not.

Considering this information, it's crucial to refrain from utilizing spatial data with deviant patterns for data recovery. Doing so may disrupt pattern inspection and lead to inaccurate determinations. The proposed algorithm aims to identify nodes most similar to the failed node (i.e., the node with missing data), ensuring the use of comparable sensor data. To assess the similarity between two time series, the NSF employs the Pearson correlation coefficient. A correlation coefficient exceeding 0.5 indicates a robust positive relationship, signifying that both variables tend to increase simultaneously. A coefficient ranging from 0.3 to 0.5 suggests a moderate positive relationship, while a coefficient between 0 and 0.3 signifies a weak positive relationship. A coefficient of 0 denotes no relationship between the variables.

For negative relationships, a coefficient between 0 and -0.3 implies a weak negative relationship, and a coefficient between -0.3 and -0.5 indicates a moderate negative relationship. A coefficient less than -0.5 indicates a strong negative relationship, where one variable tends to decrease as the other increases. The correlation coefficient provides a concise understanding of the associations between variables, serving as a descriptive

statistic to offer a high-level overview of dataset features. Utilizing the Pearson correlation coefficient, the similarity of two time-series data, denoted as A and B, is determined through the equation 3.2.

$$\rho\,(A,B) = \frac{\mu(A,B) - \mu(A)\mu(B)}{\sqrt{\mu(A^2) - \mu^2(A)}\sqrt{\mu(B^2) - \mu^2(B)}} \tag{3.2}$$

where $\mu(A)$ and $\mu(B)$ are the means of sets A and B, respectively. The spatial information is utilized as input for the IMD-MP algorithm.

The proposed approach utilized a threshold of 0.3 to determine the most closely related sensor nodes to the failed node in both the Intel lab dataset and the live dataset. Prior to applying Pearson correlation to the failed node, the missing data of this node is replaced with the mode of that data in order to avoid inconsistent correlation values caused by missing data.

Algorithm 2 Node Similarity Finding(NSF)

INPUT: Sensor groups G[][](from algorithm 1), The failed node Z, and a threshold λ
OUTPUT: Set of similar nodes SF[] of failure node Z.

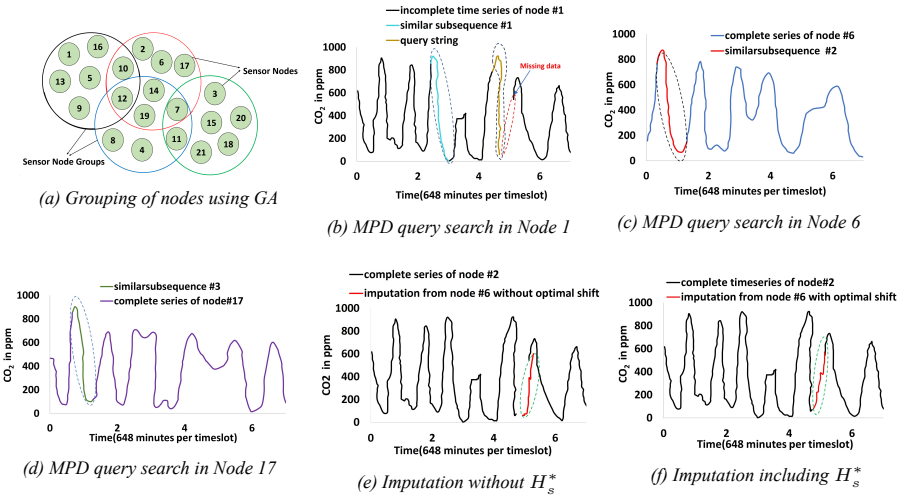
```
Begin
if Z then in G then
    SS=G[Z]
else
    for i do in G
        for j do in G[i]
            if G[i][j]==Z then
                SS.append(G[i])
            end if
        end for
    end for
end if
SF = ϕ
for (i = 1: SS.length) do
    correlation=ρ(Z, SS[i])
    if (correlation > λ) then
        SF= SF ∪ SS[i]
    end if
end for
End
```

3.2.1 Workflow of NSF

1. The algorithm checks if the failed node Z is directly present in the sensor group G. If so, it retrieves the associated group of nodes and stores it in the variable SS.

2. If the failed node Z is not directly in G, the algorithm searches for the node Z within all the groups G[i] in G. When it finds a group G[i] that contains the node Z, it appends that group to the list SS. This step gathers all the groups that have a node similar to the failed node Z.

3. The variable SF is initialized as an empty set. This set will store the nodes that are found to be similar to the failed node Z.



(a) Grouping of nodes using GA

(b) MPD query search in Node 1

(c) MPD query search in Node 6

(d) MPD query search in Node 17

(e) Imputation without H_s^*

(f) Imputation including H_s^*

Figure 3: Imputation process using proposed method

4. The algorithm then iterates through the elements of SS (groups of nodes that contain nodes similar to Z).
5. For each node SS[i] in SS, the Pearson correlation coefficient ρ(Z, SS[i]) is computed, representing the correlation between the failed node Z and each node SS[i].
6. If the Pearson correlation coefficient between Z and a node SS[i] is greater than the threshold λ, it means that node SS[i] is sufficiently similar to the failed node Z in terms of linear correlation. In that case, the node SS[i] is added to the set SF, indicating that it is a similar node to the failed node Z.
7. After completing the iteration, the algorithm returns the set of similar nodes SF[], which consists of nodes that exhibit a significant linear correlation with the failed node Z according to the threshold λ and the Pearson correlation coefficient.

After locating similar sensor nodes,the data from these nodes will be used as input for IMD-MP to impute missing values using MPD and the data pattern with the highest degree of similarity. Here is one important point that needs to be made clear: NSF (Nearest Sensor Fusion) is used to find the similarity between neighboring sensor data, whereas the IMD-MP algorithm utilizes a similarity score to identify the most similar subsequence to the preceding missing subsequence, also known as the query string.

3.3 Reconstruction of Missing Data using MPD

This is the third phase of the proposed method. MPD plays a vital role in this phase. The Matrix Profile is a powerful technique used in time series data analysis for efficient and accurate similarity searches. It is primarily used for finding repeated patterns or motifs within a time series data set. The basic principle of the Matrix Profile distance for query search can be summarized as follows:

- Sliding Window Sub-sequence Comparison: The first step in the Matrix Profile distance calculation involves comparing a fixed-length sub-sequence (or query) with all possible sub-sequences of the same length in the time series data. This comparison is typically performed using a distance metric such as Euclidean distance or Pearson correlation.
- Matrix Profile Computation: The Matrix Profile is a new time series containing the minimum distances (or similarity scores) for each sub-sequence in the data compared to the query. It represents how well each sub-sequence matches the query.
- Finding the Minimum: The Matrix Profile enables us to identify the most similar subsequence in the time series data to the query by locating the minimum value in the Matrix Profile. This minimum value indicates the best match or the motif in the data that most closely resembles the query.
- Locating Motifs: Once the minimum value is found, its corresponding index in the time series data indicates the starting position of the best-matching subsequence. This position helps identify the motif in the data that matches the query.
- Lower Bounds: In practice, the Matrix Profile distance search can be computationally expensive, especially for large data sets. To speed up the process, lower bounds (e.g., STAMP, STOMP, etc.) can be utilized to reduce the number of distance calculations and prune non-promising candidates, making the search more efficient.

The IMD-MP finds the imputation sub-sequence in sequence of steps. The basic inputs for this IMD-MP algorithm are the failure node's(Z) incomplete time series along with time series data of its similar nodes. By employing the Matrix Profile distance search, one can quickly and accurately identify repeated patterns or motifs in time series data, which is valuable for various applications, such as anomaly detection, pattern recognition, and data mining in various domains including finance, healthcare, and industrial process monitoring.

Table 1 presents some of the most essential notations and definitions for uni-variate time series. The overall imputation processes using IMD-MP can be observed in Figure 3. The imputation process is presented in algorithm 3.

3.3.1 Workflow of IMD-MP algorithm

1. The algorithm takes several inputs, including the incomplete time series K for the failure node Z, the similarity node set $SF[Z]$ obtained from Algorithm 2 , a weighting factor β, the length of MPD-subsequence q, and the number of similar sub-sequences x. The goal is to impute the missing data and obtain the imputed time series K_imp for the failed node Z.
2. The algorithm begins by constructing a missing sequence $MS[j, m]$ and a query $QS[j, n]$. It sets the values in $K[j - n : j - 1]$ to NA to represent the missing data, and $MS[j, m]$ is assigned the values from $K[j : j + m - 1]$.
3. The algorithm defines $MF[j, m]$ as a modified missing sequence, and $QS[j, n]$ as the query string. These steps help prepare the data for further processing.
4. The algorithm initializes the variable $score$ to infinity.

5. It initializes a temporary score variable, $temp_{\text{score}}$, to 0.
6. The algorithm enters a loop that iterates over each node S in the similarity node set SF.
7. For each S, it generates the All Sequence Set (ASS) by moving a window of size w with a step size of b from S. This ASS contains all possible sub-sequences of S.
8. The algorithm calculates the Minimum Pairwise Distance (MPD) between the query $QS[j, n]$ and every sub-sequence in the ASS. This step results in a distance-profile (DP).
9. From the sorted DP, the algorithm creates a Top-Index Set (TIS) of length x by selecting the x least indices. The indices in TIS are denoted as f_x, and the corresponding DP values are denoted as dp_f.
10. The algorithm checks if the DP values in TIS are present in the range R. If not, it replaces the f_i value with $(argmax(DP) - 1)b + 1$ and sets dp_{f_i} as the maximum value in DP.
11. Next, the algorithm obtains the Top-Similar Subsequence Set (TSS_x) by selecting the x most similar subsequences from the ASS.
12. For each subsequence in TSS_x, the algorithm checks if the corresponding imputation subsequence ${TSS'}_x$ is present in the set of real numbers ${R'}_m$. If not, it removes the subsequence from both ${TSS'}_x$ and TSS_x. Otherwise, it calculates the optimal shifting H_s^* using the provided equation.
13. The algorithm calculates the similarity score (ss) between each subsequence $K_{f_s,w}$ in TSS and the query QS[j,n] using the given similarity score equation.
14. The algorithm selects the following sub-sequence $K_{f'_s,m}$ from $K_{f_s,w}$ with the smallest similarity score ss_k.
15. The missing sequence $MS[j, m]$ is updated to be the selected imputation subsequence $K_{f'_s,m}$ shifted by the optimal shifting value H_s^*.
16. The temporary score variable $temp_{\text{score}}$ is updated with the similarity score ss_k.
17. The above steps are repeated for each node S in the similarity node set SF, and the algorithm keeps track of the minimum similarity score and its corresponding modified missing sequence $MF[j, m]$.
18. Finally, the algorithm outputs the imputed time series K_imp by replacing the missing gaps in the original time series K with the modified missing sequence $MF[j, m]$.

The working of IMD-MP with an example can be seen in Figure 3. The node 2 with sensor 2 has the missing sub-sequence in Figure 3b for CO_2, query string also marked.After applying NSF to similar nodes predicted similar nodes were node 6 and node 17 which were having similar sub-sequences to the query string as shown in Figures 3c and 3d respectively. After applying MPD the sub-sequence present in node 6 has least similarity score value after shifted to optimal height in figure 3e. This sequence will be used as imputation sequence in node 2 and the final imputed sequence is present in Figure 3f.

S.No	Notation	Explanation
1	K	An uni-variate time series of length 'T' with a sequence of real valued time series such as K = $k_1, k_2, ..., k_T$
2	$K_{j,w}$	$K_{j,w} = k_j, k_{j+1}, ..., k_{j+w-1}$ is a sequence of w consecutive values in K starting from the j th position,where $1 \leq j \leq T - l + 1$.
3	$MS_{j,m}$	In the K(Uni-variate series), a missing sub-sequence is a continuous subset of incomplete data from position j to position $j + m - 1$ with a value of NA
4	$QS_{j,n}$	A query-string with 'n' length that precedes a missing sub-sequence;such that $QS_{j,n}$ = K[j-n : j-1].
5	$S_{w,b}$	Sliding window: By moving a window with size w of step-length b, one can discover all required sub-sequences for a provided time series.
6	ASS	All possible sub-sequences set in uni-variate series K, obtained by moving a window of size w with a leap size b; the total number of sub-sequences is $\lfloor \frac{T-w}{b} + 1 \rfloor * no.of similar nodes$; ASS= $\{K_{1,w}, K_{1+b,w+b}, \ldots..\}$
7	DP	A set of MPD values calculated in between query $QS_{j,n}$ and every sub-sequence in the set of all ASS of univariate time series K called as distance-profile (DP).
8	TIS_x	A Topmost-Index Series (TIS_x) of length x can be taken from DP by sorting in increasing order;$TIS_x = \{f_x \mid f_x \in G, 1 \leq x \leq \lfloor \frac{T-w}{b} + 1 \rfloor \}$, where f_x is the x-th least index of DP after sorting; $dp_{f_i} \leq dp_{f_{i+1}}$, where 1 $\leq i \leq x - 1$;
9	TSS_x	A Top most-Similar Sub-sequence Series (TSS_x) of K is a set of x sub-sequences that are most similar to a query $QS_{j,n}$; $TSS_x = \{K_{f_1,w}, K_{f_2,w}, \ldots.., K_{f_k,w}\}$
10	TSS'_x	An Imputation Sub-sequence Set TSS'_x is composed of the following sub-sequence of $K_{f_s,w}$, in a Top-Similar Sub-sequence Set TSS_x; $(TSS'_x) = \left\{K_{f'_1,m}, K_{f'_2,m}, \ldots.., K_{f'_k,m}\right\}$, where $1 \leq s \leq x$ and $f'_f = f_s + w$; All indices in (TSS'_x) are termed as imputation sub-sequences.
11	H^*_s	Optimal height value to which the imputation sub-sequence should be shifted.The value of H^*_s can be calculated by using the equation 3.3
12	ss	The similarity between each sub-sequence $K_{f_{s,w}}$ in TSS and $QS_{j,n}$ for missing data imputation is represented by a value called the similarity score (ss), which can be calculated using equation 3.7

Table 1: Terms used in IMD-MP algorithm

Due to the MPD measure provides advantages over traditional distance measures like Euclidean distance and Dynamic Time Warping (DTW) this research utilized it in IMD-MP. The advantages are listed here.

1. Permits comparisons between time series of various lengths: Unlike Euclidean distance and DTW, which require time series of the same length for direct comparison, MPD allows for comparisons between time series of different lengths. This is a significant advantage when dealing with real-world datasets where time series can have varying lengths due to missing data or different sampling rates.

2. Resistant to various data irregularities: MPD is more robust to data irregularities like surges, failures, wandering baselines, and missing values. This robustness is crucial when working with real-world datasets that often contain noise, anomalies, and gaps in the data.

3. Additional in-variances: MPD not only offers amplitude and offset in-variances like DTW and Euclidean distance but also provides phase, order, linear trend, and stutter in-variances. These additional in-variances make MPD more flexible and capable of handling complex temporal patterns in time series data.

4. Effectiveness with absent data: Whether the absent data is implicit, explicit, or a combination of both, MPD is unaffected. This characteristic ensures that the distance measure remains reliable and consistent even when dealing with incomplete time series data.

5. Single parameter requirement: MPD only requires a single parameter, the sliding window size, which is relatively easy to understand and tune. In contrast, other distance measures like DTW may require more complex parameter settings.

6. Amortised sub-sequence search cost: Although a single comparison using MPD may be computationally costly compared to Euclidean distance, the amortised cost of sub-sequence search (e.g., in time series similarity search tasks) is comparable to Euclidean distance. This means that when analyzing large datasets or performing similarity search operations, MPD remains efficient.

Overall, the advantages of MPD make it a compelling choice for analyzing and comparing time series data, particularly in real-world scenarios where data irregularities, missing values, and varying time series lengths are common challenges. Its ability to handle a wide range of invariances and its ease of use with a single parameter further contribute to its superiority over traditional distance measures like Euclidean distance and DTW in various time series analysis tasks.

3.4 Optimal H_s^* value and Similarity Score

3.4.1 Calculating H_s^* value

To maintain imputation accuracy after getting the most similar subsequence to the query string, the temporal correlation can be obtained by shifting that most similar subsequence to optimal height. This can be calculated by the equation 3.3.

$$H_s^* = min\left(v_1^t,\ max\left(\frac{k_{j-1} - k_{f_s'} + k_{j+m} - k_{f_s'+m-1}}{2}\ ,\ v_2^t\right)\right) \tag{3.3}$$

$$R_{\max} \geq \left(H_s^* + K_{f'_s,m}\right) \geq R_{\min} \tag{3.4}$$

And also

$$v_1^t = R_{max} - min\left(K_{f'_s,m}\right) \tag{3.5}$$

$$v_2^t = R_{min} - min\left(K_{f'_s,m}\right) \tag{3.6}$$

where R_{min} and R_{max} are the measuring ranges of sensor.Here, to guarantee the minimum imputation subsequence value within the [R_{min} , R_{max}],the constraint S3.4 must be considered. The values of v_1^t and v_2^t can be calculated by equations 3.5 and 3.6.

3.4.2 Calculating Similarity Score value

The similarity score is based on euclidean distances. If the $K_{f_s,w}$ is similar to $QS_{j,n}$ then no need of optimal shifting. Otherwise, $K_{f_p,w}$ is optimally shifted with H_s^*. To consider both these cases we have used weighting factor β with a value of 0.5. This Similarity Score gives the similarity between each $K_{f_{s,w}}$ in TIS_x and query string $QS_{j,n}$ for imputation of missing data. The similarity score can be calculated by $K_{f'_s}$ by adding H_s^* using 3.7. Where the values of a and b can be calculated by the equation 3.8, 3.9 respectively.

$$SimilarityScore(ss) = \beta\sqrt{\frac{1}{n}}||K_{f'_s,w} - H_s^* - QS_{j,n}||_2 + (1-\beta)\sqrt{\frac{a^2+b^2}{2}} \tag{3.7}$$

$$a = K_{j-1} - K_{f'_s} - H_s^* \tag{3.8}$$

$$b = K_{j+m} - K_{f'_{s+m-1}-H_s^*} \tag{3.9}$$

A subsequnce that has the smallest similarity score can be considered as most similar sequence of $QS_{j,n}$ which is used to impute the data.The section 5 briefly explains the evaluation and result analysis in detail.

4 Evaluation and Result Analysis

To evaluate the performance of our proposed method, this research has implemented with the two cases of evaluation set-up. Also the performance improvement of the proposed model has been evaluated and proved by comparing with exiting imputation models. This section further organized as experimental setup, evaluation metrics and performance evaluation with compared to recent existing imputation algorithms such as Itr-MS-STLecImp (Itr-MS-STL) [Liu et al. 2020], VTN [Deng et al. 2022], DLIN [Hallaji et al. 2021] and DTWBI [Caillault et al. 2020].

Algorithm 3 Imputation of Missing Data using MP-distance(IMD-MP)

INPUT: $K = \{k_1, k_2, k_3, \ldots\ldots, k\}$: incomplete time series of failure node Z,
SF[Z] :Similarity node set of Z obtained from alg 2 with their data.
β: weighting factor, q: MPD-subsequence length
x: the number of similar subsequences
OUTPUT:The Imputed time series K_{imp} of failed node Z

Construct a missing sequence $MS_{j,m}$and a query $QS_{j,n}$
$K[j-n:j-1] \leftarrow NA$
$MS_{j,m} \leftarrow K[j:j+m-1]$
$MF_{j,m} \leftarrow K[j:j+m-1], QS_{j,n} \leftarrow K[j-n:j-1]$
score=∞
$temp_{score}$=0
for each S in SF **do**
    Getting All sequence Set (ASS) by moving a window size of w with step size b from S
    $$ASS = \left\{ K_{1,w},\ K_{1+b,w}, \ldots, K_{1+\lfloor \frac{T-w}{b} \rfloor b,w} \right\}$$
    Compute MPD_q between the query $QS_{j,n}$ and every sub-sequence ASS and obtain a distance-profile
    $$DP = \left\{ dp_f | \ f \in \left\{ 1, 1+b, \ldots..1+\lfloor \frac{T-w}{b} \rfloor b \right\} \right\}$$
    A Top-Index Set (TIS_x) of length x can be taken from sorted DP in increasing order; $TIS_x = \left\{ f_x \right| \ f_x \in Z, 1 \leq x \leq \left\lfloor \frac{T-w}{b} + 1 \right\rfloor \ \}$, where f_x denotes the x-th least index of DP after sorting; $dp_{f_i} \leq dp_{f_{i+1}}$, where $1 \leq i \leq x-1$;
    for ($f_i \in TIS_x$) **do**
        if ($dp_{f_i} \notin R$) **then**
            $f_i \leftarrow ((argmax(DP)-1)b+1)$ and $dp_{f_i} \leftarrow max(DP)$
        end if
    end for
    Obtain a Top-Similar Sub-sequence Set (TSS_x);
    $TSS_x = \{K_{f_1,w}, K_{f_2,w}, \ldots, K_{f_x,w}\}$
    Get an Imputation Sub-sequence Set ($TSS\prime_x$) and compute a similarity score(ss);
    for (s= 1:x) **do**
        if $\left\{K_{f'_s,m}\right\} \notin \mathbb{R}^m$) **then**
            $TSS'_x \leftarrow TSS'_x \setminus \left\{K_{f'_s,m}\right\}$ and$TSS_x \leftarrow TSS_x \setminus \{K_{f_s,w}\}$
        else
            Calculate the optimal shifting H^*_s using 3.3.
            Get the Similarity Score(ss) using 3.7.
        end if
    end for
    Obtain a following sub-sequence $K_{f'_s,m}$ of $K_{f_s,w}$ with the smallest similarity score ss_k.
    $MS_{j,m}$=$K_{f'_s,m}$+H^*_s
    $temp_{score}$=ss_k
end for
if $temp_{score}$<score **then**
    $MF_{j,m} \leftarrow MS_{j,m}$
    score=$temp_{score}$
else
    no change in $MF_{j,m}$
end if
Output the Imputed time series K_{imp} by replacing missing gap with $MF_{j,m}$

Name	Specication
Arduino IoT Module	Xtensa dual-core 32-bit
LoRa module	RF96 (300kbps) with 865-867 Mhz
Wi-Fi module	802.11n (150mbps) with 2412-2484 MHz
Bluetooth module	BLE V4.2 BR-EDR
DHT11	Temperature Range: 0°C to 50°C with accuracy around ±2°C, Humidity Range: 20% to 90% with accuracy ±5% RH
MQ135	Ammonia, nitrogen oxides, alcohol, CO2 typically covers 10ppm to 1000ppm
Robodo-130008	Rainwater sensor-Measures the presence of rainwater
Robo-ZX-FLMS-01	Flame Sensor-Usually sensitive to IR and UV light emitted by flames
111-AQI	Sensitive to a range of gases like Ammonia, Nitrogen oxides, Benzene, Alcohol ,Carbon monoxide ,Carbon dioxide.

Table 2: Sensor Node Specification

4.1 Experimental setup

As mentioned, the proposed method evaluated on two different datasets. In the case-1 Intel lab dataset(for the purpose of evaluating our proposed method with benchmark dataset) is used and in the case-2, a campus specific IoT network has been implemented and which generates a live dataset with edge computing configured as a IoT-testbed for environmental monitoring in VIT-AP University, India.

Intel Berkeley Research Laboratory (IBRL) dataset: Experimental setup is not required for the IBRL dataset. Intel's Berkeley Research Lab established a network of wireless sensors consisting of 54 nodes to gather data on the surrounding area of the lab. The nodes were equipped with a range of sensors, including those for humidity, light, temperature, and voltage. These sensors were used to collect data on various environmental conditions at regular intervals of 31 sec from 28-02-2004 to 5-04-2004, resulting in a total of 2.3 million readings and 4300 recordings. The sensor data that was gathered has been made accessible for research in a public repository. In IBRL dataset, the proposed method was applied on temperature, humidity and light features only. So,the voltage values from IBRL dataset were removed. The following section, we discussed about the Live AQS dataset.

Air Quality monitoring System(AQS) Live dataset: A campus-specific IoT network has been established in VIT-AP University, India. This network is designed to gather real-time data and includes an edge computing server that is configured as an IoT test-bed for environmental monitoring. Within the smart campus project at VIT-AP University, 2 separate IoT networks have been implemented as an Air Quality monitoring System and an Environmental Monitoring System. These networks are managed through the edge computing server located at the Intel IoT Center for Excellence lab. The AQS project has been regarded as the verification dataset in this context. The AQS system was implemented by deploying twenty Arduino-based IoT nodes across the campus. Every node was securely positioned and linked to the campus Wi-Fi network. The node

comprises sensors such as DHT11, MQ135, Rainwater sensor, and flame sensor, as specified in Table 2. Every node was set up as a time-driven application, sending data to the edge server located at the IoT CoE center every 2 minutes. In addition to the IoT sensors installed in the AQS project, we have incorporated an additional MG111-AQI sensor at every node in order to assess the effectiveness of the proposed system.

4.2 Evaluation Metrics

WMAPE, NMAE, and Similarity Percentage metrics were used to evaluate IMD-MP on the two datasets. i.e., IBRL dataset and AQS live dataset. This subsection listed the metrics used for evaluating the proposed method. The three metrics used to evaluate were Similarity Percentage, Weighted Mean Absolute Percentage Error (WMAPE) and Normalized Mean Absolute Error (NMAE).

4.2.1 Similarity

Similarity(IM,AD) reflects the similarity between the actual data (AD) and the imputed data (IM). It is calculated by 4.1:

$$Similarity(IM, AD) = \frac{100}{M} \sum_{i=1}^{M} \frac{1}{1 + \frac{|IM_i - AD_i|}{max(AD) - min(AD)}} \tag{4.1}$$

In the given context, M signifies the number of missing values. A higher similarity value, within the range of 0 to 100, suggests that a technique possesses superior capabilities for reconstructing missing values.

4.2.2 Weighted Mean Absolute Percentage Error (WMAPE)

WMAPE, is a modified version of MAPE that integrates a weighted arithmetic mean based on mean absolute percent errors. In this approach, absolute percent errors are typically weighted by actual values, addressing the issue of infinite error and providing a more accurate assessment of the model's performance. The formula for calculating WMAPE is given by Equation 4.2:

$$WMAPE = \frac{\sum_{i=1}^{n} |AD_i - IM_i|}{\sum_{i=1}^{n} |\mathbf{AD_i}|} \tag{4.2}$$

4.2.3 Normalized Mean Absolute Error(NMAE)

Assesses the accuracy of an imputed value relative to the corresponding true value time series and is computed as given in Equation 4.3:

$$NMAE = \frac{1}{M} \sum_{i=1}^{T} \frac{|IM_i - AD_i|}{AD_{max} - AD_{min}} \tag{4.3}$$

4.3 Datasets and Pre processing

The first dataset on which the proposed method evaluated was Intel's Berkeley Research Lab (IBRL) dataset and the second one was AQS (Air-Quality System) live dataset. The real-time data was gathered at the edge computing server obtained from MG111-AQI was utilized by IMD-MP (CO_2, NH_3, and C_6H_6). The accuracy of data imputation was then confirmed using MQ135 sensor data. The data had been collected over the period from 31^{st} July 2022 to 1^{st} December 2022. The total 64,800 (90x24x30 i.e, 90 days,each day 24 hours and for each hour 30 readings) records were collected at each sensor node. Before applying these two datasets to the experiments, these datasets should be pre processed. Any missing values present in these datasets were replaced by mean imputation before applying IMD-MP.

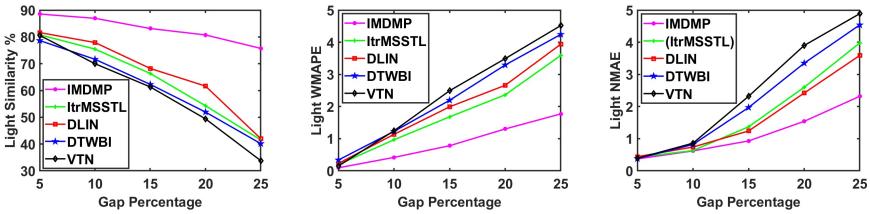
Parameters	Values
Missing gaps	5%, 10%, 15%, 20% and 25%
Query length w.r.t missing gap	0.25, 0.5, 1.0, 2.0, and 4.0
Weighting factor	0.75
Sliding Window step size	0.2 of query length
Number of similar sub-sequences	5

Table 3: Experimental parameter values for IMD-MP algorithm

4.4 Evaluation on the IBRL dataset

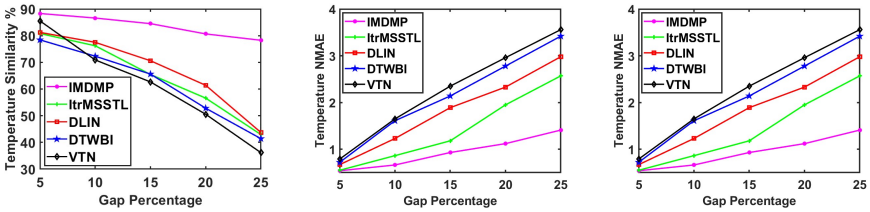
With a query rate of 0.25, 0.05, 1.0, 2.0, and 4.0, we have evaluated the IBRL dataset. To evaluate IMD-MP on the IBRL dataset, the first grouping algorithm (GA) should be applied. To apply the grouping algorithm, the neighbors of all sensors should be identified. For the Intel Lab dataset, the x and y coordinates of each sensor node were provided on the website. By taking those coordinates, the Euclidian distance was calculated between each sensor and stored. After storing these values, they were sorted. For every sensor, the neighbors were obtained by considering the top 5 nearest neighbors, i.e., the sensors that have a shorter Euclidian distance. Here, the reason for selecting the top 5 neighbors is to reduce the time complexity of the proposed method because if the number of neighbors is greater, then there may be a chance of an increase in similar sensor nodes in NSF, which will further increase the number of sub-sequences in ASS in IMD-MP, resulting in increased time complexity.

As previously mentioned, in the imputation process on the Intel dataset, we considered humidity, temperature, and light data. The Intel dataset consists of homogeneous data from 54 sensor nodes, resulting in the formation of 9 groups. One of the groups formed by the grouping algorithm contained sensor nodes 1, 2, 4, 3, 33, 34, and 35. We purposefully created missing gaps in data of node 1 for evaluating IMD-MP with 5%, 10%, 15%, 20%, and 25% of lengths at various time stamps. Secondly, the NSF algorithm was applied to the data of 1, 2, 3, 4, 33, 34, and 35 to find the most similar sensor nodes to node 1. The resulting Pearson's correlation coefficients of node 1 temperature sensor data with 2, 3, 4, 33, and 34 were 0.281, 0.3654, 0.2613, 04217, and 0.325. We considered the threshold λ with a 0.3 value in the NSF algorithm. The most similar nodes for node 1



(a) Gap % vs Similarity (%) *(b) Gap % vs WMAPE* *(c) Gap % vs NMAE*

Figure 4: Light data



(a) Gap % vs Similarity (%) *(b) Gap % vs WMAPE* *(c) Gap % vs NMAE*

Figure 5: Temperature data

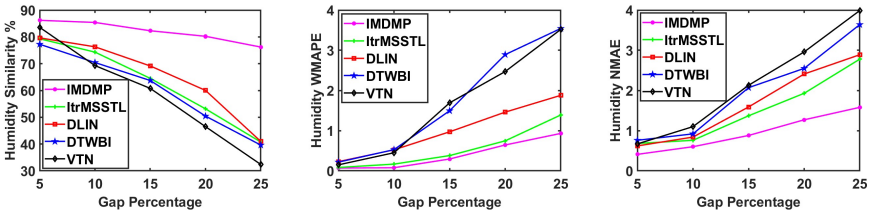
were selected as 3, 33, and 34. Similarly, for humidity and light data, the same set of nodes data was utilized to impute the missing data from Node 1.

Finally, algorithm 3 (IMD-MP) was used to impute the missing sequence of Node 1. The parameters used in the IMD-MP algorithm are mentioned in Table 3. With varying query rates at all missing rates, IMD-MP was implemented on light, temperature, and humidity data. The baseline models were used to compare the evaluation outcomes. Assuming a query rate of 2.0, Figures 4a, 5a and 6a , show the percentage of similarity between the original data and the data that was reconstructed for light, temperature, and humidity, respectively. Similarly, the WMAPE values were depicted in Figures 4b, 5b and 6b. The corresponding NMAE values were depicted in Figures 4c, 5c and 6c.

4.4.1 Evaluation with live dataset

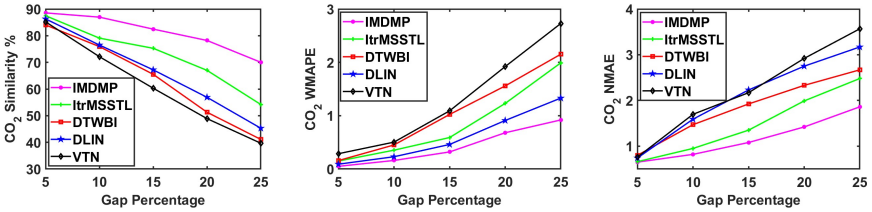
As mentioned before, we configured an IoT test bed at our Intel IoT CoE. In this case, the MG111-AQI sensor data CO_2, NH_3, and C_6H_6 were observed from each IoT node from 31^{st} July 2022 to 29^{th} October 2022 and were particularly marked to be used for imputation performance analysis. In order to validate IMD-MP with complete data, the data from the other sensor, MQ135, which was attached to the same IoT node, was used. The 20 nodes intended for data collection were grouped using the algorithm GA into four different groups. The process of grouping was done based on the neighboring nodes of each node. The neighboring nodes were obtained by calculating the Euclidian distance based on the location information of all nodes. After applying GA based on the list of neighbors of all nodes, the nodes 2, 6, 10, 12, and 17 were formed as one group. Then the proposed model was implemented on the sensor node-2's data taken from the edge computing server.

The next step in the proposed method was identifying similar nodes to failed node using the NSF algorithm. Here, Node 2 was considered a failed node; the corresponding similar nodes considered were 6, 12, and 17 because the Pearson's correlation values of



(a) Gap % vs Similarity (%) (b) Gap % vs WMAPE (c) Gap % vs NMAE

Figure 6: Humidity data



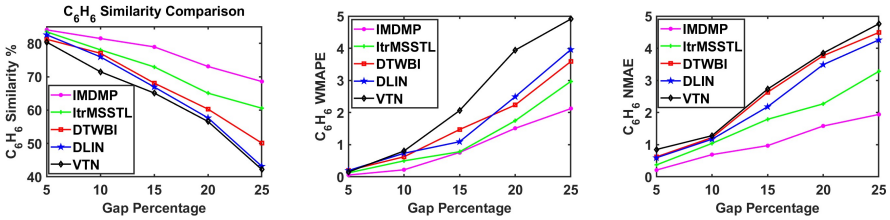
(a) Gap % vs Similarity (%) (b) Gap % vs WMAPE (c) Gap % vs NMAE

Figure 7: CO_2 data

these nodes when compared against Node 2 were greater than 0.3 in the case of CO_2. In the case of C_6H_6 and NH_3 data, the similar nodes considered were 6, 12, and 17. We have evaluated algorithm 3 on these nodes data with parameters specified in Table 3. The missing gaps created were 5%, 10%, 15%, 20%, and 25% (i.e., 4.5 days, 9 days, 13.5 days, 18 days, and 22.5 days). All the missing gaps were imputed with 0.25, 0.5, 1.0, 2.0, and 4.0 query rates with similar substances of 5). Figure 3 demonstrates the imputation process using IMD-MP. As shown in Figure3a, groups were formed by nodes present in the AQS project. Figure 3b contains the node 2 CO_2 data where the missing data has a 5% (4.5 days) missing gap. The total records we gathered were 64,800 of the 90-day data. This 90-day data was divided into 10 slots. Each slot contains 648 records, i.e., 9 days of data. The query rate was 1.25 of the missing sub-sequence length. This query string is also marked in Figure 3b. To find the imputation sub-sequence for the missing gap filling, similar sub-sequences were searched over node 2, node 6, and node 17 using IMD-MP. The corresponding matching sequences with query strings were identified and marked in Figures 3b, 3c, and 3d as similar sub-sequences 1, 2, and 3, respectively. Similarly, another two sequences are also present in node 6 and node 17. The sub-sequence 3 at node 17 had the least MPD and high similarity, which was selected as the imputation sub-sequence as shown in Figure 3f. The sub-sequence was optimally shifted and imputed at sensor 2's missing gap, as shown in Figure 3f. Figures 7a, 8a, and 9a show the percentage of similarity between the original data and the reconstructed data of CO_2, NH_3, and C_6H_6. Figures 7b, 8b, and 9b show the WMAPE and Figures 7c, 8c, and 9c show the NMAE.

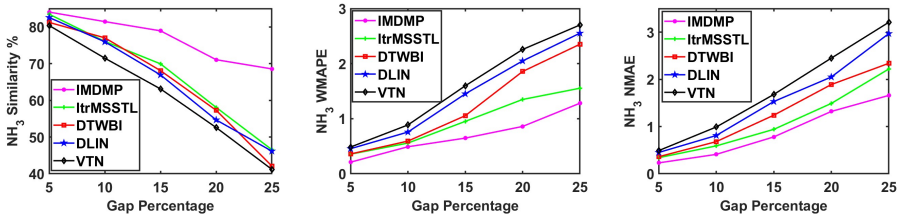
5 Discussion

This section highlighted the key performance of IMD-MP over the remaining methods. By observing Figures 7, 8, and 9, as the missing gap increased, the percentage of similarity decreased for IMD-MP to 5.44%, 10.23%, 16.63%, and 24.68% w.r.t. missing gaps of



(a) Gap % vs Similarity (%) (b) Gap % vs WMAPE (c) Gap % vs NMAE

Figure 8: C_6H_6 data



(a) Gap % vs Similarity (%) (b) Gap (%) vs WMAPE (c) Gap % vs NMAE

Figure 9: NH_3 data

10%, 15%, 20%, and 25% in the case of temperature data. The decrease in similarity percentage of ItrMSSTL on temperature data was 6.96%., 13.61%, 20.00% and 34.08% w.r.t. missing gaps of 10%,15%,20%, and 25% missing gaps, respectively. This is because the ItrMSSTL was an iterative imputation approach; in each iteration, the imputation performance depends on the previous imputed data only. If one iteration fails to impute part of the missing gap accurately, that affects the remaining parts imputation performance as well.

Similarly, when compared with all the base lines, IMD-MP imputed the data with greater similarity values of 34.56, 36.87, and 42.13 over DLIN, DTWBI and VTN, respectively, at a missing gap of 25% in temperature with a query rate of 2.0. Figures 6a, 6b and 6c depicted that the performance of all methods except IMD-MP decreased in terms of similarity and increased in terms of WMAPE and NMAE w.r.t. an increase in missing

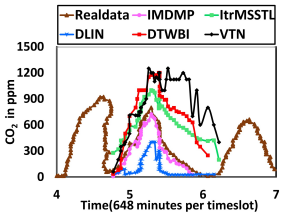


Figure 10: CO_2 imputation with 10% missing gap and 2.0 query rate

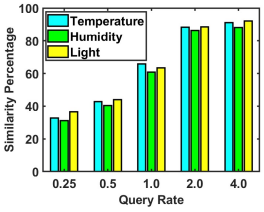


Figure 11: Comparison on Similarity (%) with varying query rate of Intel dataset

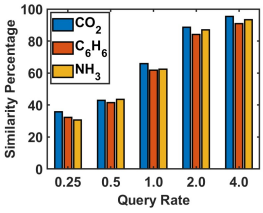


Figure 12: Comparison on Similarity (%) with varying query rate of Live dataset

gap lengths. When compared the similarity percentage of IMD-MP with temperature and humidity data,the humidity data has lower similarity values than temperature because humidity data has less seasonality than temperature. Figures 4a, 4b, and 4c show the similarity of light data in the IBRL dataset by methods. Notably, ItrMSSTL has more errors than IMD-MP, even though it did well. This is because the performance of each iteration depends on how well the previous iteration did with imputation. DTWBI used Euclidean distance to figure out imputation sequence, its error rate went up as the gap percentage went up because noise affected its performance too. At a missing of 5%, VTN performed well, but its performance decreased with an increase in missing gap (10%, 15%, 20%, and 25%) length because VTN imputation depends on the number of temporal neighbors, which will become less if gap length increases.

For the AQS live dataset, in the case of CO_2 data, the percentage decline in similarity of IMD-MP w.r.t. missing gaps from 5% to 25% was 1.79%, 6.76%, 11.66%, and 21.01%, which were very less comparable with ItrMSSTL.(14.17%, 24.25%, 38.19%, and 51.83%), DTWBI (11.68%, 21.24%, 36.86% and 52.76%) DLIN (10.95%, 23.84%, 40.29%, and 54.53%) and VTN (15.32%, 29.07%, 42.61%, and 58.12%). As shown in Figures 8a, 8b, and 8c, in the case of C_6H_6, the performance of all methods over the remaining data features (temperature, humidity, light, CO_2, and NH_3) increased in terms of similarity because C_6H_6 has more seasonality compared with other features. Figure 10 provides a visual comparison of the imputed values from IMD-MP for NH_3 data from time slot 5 to time slot 6 with a missing gap of 15% and other approaches with the actual values. It can be observed that IMD-MP's imputed values have a pattern shape that is more similar to the true values. This lends credence to the idea that when the missing gap is sufficiently significant, IMD-MP provides greater performance. The time series with a strong seasonality component, such as the NH_3 dataset, appears to be unproductive with techniques such as ItrMSSTL, DLIN, DTWBI, and VTN at the large missing gap, as these methods do not adequately account for the characteristics of the historical time series with spatial correlations. In these numbers, the imputed values from ItrMSSTL, DLIN, DTWBI, and VTN all generate upward and downward trends but have a poorer fit with the actual values than IMD-MP.

5.1 Computational complexity

In our experiment, the algorithms were executed sequentially, and the output of GA is passed to NSF, and the output of NSF is passed to IMD-MP, the overall time complexity is computed by summing the individual complexities using Equation 5.1.

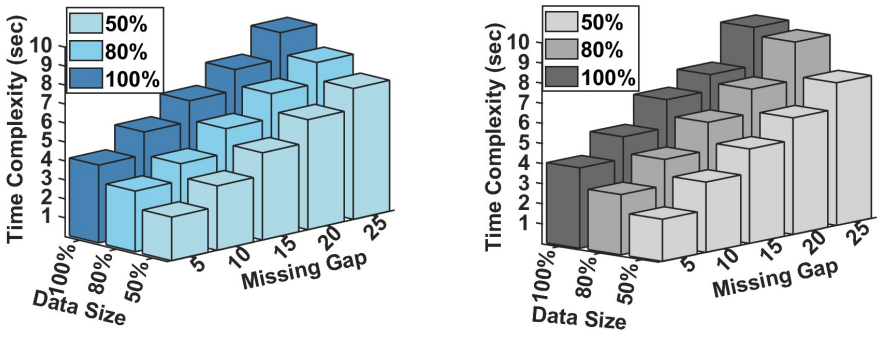
$$\mathcal{O}_t = \mathcal{O}_t^{GA} + \mathcal{O}_t^{NSF} + \mathcal{O}_t^{IMD-MP} \tag{5.1}$$

where $\mathcal{O}_t^{GA}, \mathcal{O}_t^{NSF} and \mathcal{O}_t^{IMD-MP}$ are time the complexities corresponding to GA, NSF and IMD-MP algorithms. They can be computed using the Equations 5.2, 5.3 and 5.4.

$$\mathcal{O}_t^{GA} = O(n \cdot m) \tag{5.2}$$

where n is the number of sensors, and m is the average number of neighbors for each sensor.

$$\mathcal{O}_t^{NSF} = O(|G|^2 \cdot |SS|) \tag{5.3}$$



(a) With IBRL dataset (b) With AQS Live dataset

Figure 13: Time Complexity of IMD-MP

where $|G|$ is the number of sensor groups, and $|SS|$ is the size of the similarity set SS.

$$\mathcal{O}_t^{IMD-MP} = \mathcal{O}\left(|SF| \cdot (w + x \cdot \log(x) \cdot (q + \log(q)))\right) \tag{5.4}$$

where, $|SF|$ is the size of the similarity set obtained from the NSF algorithm, w is the window size, x is the number of similar sub-sequences. q is the MPD-subsequence length.

To examine the time complexity of the proposed method, we have considered different dataset sizes like 50%, 80% and 100% at various sizes of missing gaps like 5%, 10%, 15%, 20% and 25% with a query rate of 2.0. The Figures 13a and 13b depicts the overall time complexity of IMD-MP in case of temperature data (IBRL dataset) and C_6H_6 (AQS live dataset) respectively.

When the missing gap increases from 10% to 15%, the time complexity of temperature data (IBRL dataset) in Figure 13a shows a percentage change of approximately 35.29%. Similarly, as the missing gap increases from 15 to 20, the time complexity shows a percentage change of around 26.09%, suggesting a moderate increase. From 20% to 25% in missing gap, the percentage change in time complexity is approximately 18.97%, indicating a relatively smaller increase. This indicates a substantial increase in time complexity with increase in missing gap across all data sizes. As data size increases from '50%' to '80%', the time complexity shows a percentage change of about 20.59%. This indicates a moderate increase in time complexity. When data size increases from '80%' to '100%', the time complexity shows a higher percentage change of around 29.27%.

Similarly, when the missing gap increases from 10% to 15%, the time complexity of C_6H_6 data in Figure 13b shows a percentage change of approximately 34.29%. Similarly, as the missing gap increases from 15% to 20%, the time complexity shows a percentage change of around 24.39%, suggesting a moderate increase. From 20% to 25% in missing gap, the percentage change in time complexity is approximately 22.41%, indicating a relatively smaller increase. As data size increases from '50%' to '80%', the time complexity shows a percentage change of about 23.81%. When data size increases from '80%' to '100%', the time complexity shows a higher percentage change of around 31.58%.

Overall, one can observe that larger data sizes contribute more significantly to changes in time complexity compared to smaller increments in data size. Percentage increases in time complexity are observed when missing gap or data size increases. Understanding these percentage changes provides insights into how adjustments in missing gap and data size influence the system's performance.

5.2 Parameters effecting IMD-MP

1. **Query Rate**: It is the ratio between query length and missing sub-sequence length. Several different simulations were carried out to showcase how the imputation performance of our proposed method was influenced by the query rate. This paper investigated the impact of missing sub-sequences of varying with different gap percentages (5%,10%,15%, 20% and 25%) on different query rates 0.25, 0.5, 1.0,2.0 and 4.0.

   All datasets(IBRL and AQS Live data) are analysed for the mentioned query rates and gap percentages. Specifically, Figure 11 depicts the association between Similarity Percentage and query rates for each light, temperature and humidity of Intel dataset and Figure 12 depicts the association between Similarity Percentage and query rates for each CO_2, C_6H_6, NH_3 of live dataset with a 15% missing gap. With substantial missing gaps, the query rates of 2.0 and 4.0 perform best in terms similarity Percentage. At the 15% missing gap for each dataset with 2.0 query rate decreases the similarity percentage and increases WMAPE and NMAE metrics. Figures. 11 and 12 demonstrate that a lower query rate have less capability to understand the pattern structure due to the less amount of information included in it. Thus, these results emphasise the need of choosing an optimal query rate. But if the increased query rate then it leads to increased computational time.

2. **Threshold -λ**: In NSF,the value of λ effects the computational complexity of the IMD-MP because the decreased value of λ may result in increased number of similar node which effects the number of iterations in IMD-MP.If λ is increased then it may result in decreased number of similar nodes. However, the overall computational complexity of IMD-MP depends on query rate, missing gap, λ and number of similar sub-sequences in IMD-MP algorithm values.

From the discussion, it can be noted that IMD-MP mainly had two advantages over the existing imputation methods. The first one is at a lower query rate, IMD-MP performed well because of using both spatial and temporal correlations. Even with higher missing gaps, IMD-MP got a higher similarity percentage for the imputed data. Another case where IMP-MP is superior to other methods is when the sensors exhibit more time-varying characteristics. Here, time-varying characteristics means having less seasonality within the sensors data over time. In this case also IMD-MP imputes the data by making neighboring sensors data to gather imputation sub sequences using spatial correlations. It is conceivable that the distance measure MPD used for determining similarity might have contributed to the subpar performance of the relevant algorithms. Therefore, our proposed approach surpasses state-of-the-art methods and is preferable for the imputation of time series data of IoT.

The GA, NSF, and IMD-MP algorithms, designed for operational deployment in the IoT network, exhibit high operational feasibility. User-friendliness is prioritized with intuitive interfaces, minimizing the training required for operators. Rigorous testing confirms seamless integration into operational processes, with specific adaptations for network topology and environmental dynamics. Real-time processing capabilities met stringent requirements, ensuring timely insights for operational decision-making. Robust fault tolerance is demonstrated through simulations, showcasing the algorithms' ability to gracefully handle disruptions. Overall, these connected algorithms prove to be user-friendly, adaptable, real-time, fault-tolerant, and scalable, positioning them as suitable solutions for integration into any operational sensor network.

6 Conclusion and Future work

In our study, we presented the IMD-MP method as an innovative imputation technique for uni-variate data. We assessed the effectiveness of our approach on two datasets, comprising the live dataset and the IBRL dataset. By employing three quantitative metrics (similarity percentage, WMAPE, and RMSPE), we conducted a comparative analysis of the imputation performance of IMD-MP against baseline methods. Our evaluation results presented evidence that the IMD-MP method had improved imputation performance when compared to any DTW-based imputation algorithms. IMD-MP demonstrated superior performance over the existing methods, establishing itself as a robust approach for dealing with uni-variate data characterized by significant gaps and pronounced seasonality. This superiority was attributed to the utilization of the Matrix Profile Distance (MPD) in IMD-MP, which facilitated the capture of intricate temporal and spatial patterns for the effective management of missing data.

Nonetheless, it's crucial to recognize the limitations of our approach and identify areas for future research. While IMD-MP exhibited exceptional performance in the context of uni-variate data, there is a necessity for its extension to handle multivariate time series. This expansion could involve incorporating the correlations among multiple features across various sensors, thereby enabling the application of IMD-MP in scenarios that demand simultaneous imputation of data from multiple sensors.

Further, for time series data that exhibit recurring patterns, the utilization of artificial intelligence to choose a suitable sub-sequence for imputation can significantly improve the performance of IMD-MP. By adopting this adaptive technique, IMD-MP can effectively manage a wide range of data patterns and enhance its efficacy in different circumstances. Future research should focus on optimizing the computational efficiency of IMD-MP, particularly for indexing disk-resident data. This optimization would enhance its suitability for large-scale datasets and real-time imputation tasks. Addressing the computational efficiency limitations will allow for the broader application of IMD-MP in various IoT data imputation scenarios.

Moreover, investigating techniques to approximate the triangular inequality property of MPD is crucial for ensuring accurate distance evaluations and preserving human intuitions about similarity. Approaches such as machine learning can transform MPD into a valid metric space, while indexing strategies can boost performance for disk-resident data. By addressing these areas, IMD-MP can be further improved and extended to address various IoT data imputation challenges and foster advancements in the field. To advance this proposed model, the implementation practices must be based on a live dataset that is collected from an IoT network and consists of at least 10 IoT nodes with configuration as mentioned in the Table 2.

Acknowledgment

We acknowledge the support for the evaluation of the proposed model from Intel IoT Center for Excellence, VIT-AP University.

References

[Agbo et al. 2022] Benjamin Agbo, Hussain Al-Aqrabi, Richard Hill, Tariq Alsboui (2022): "Missing data imputation in the Internet of Things sensor networks"; Future Internet, vol. 14, no. 5, pp. 143.

[Ageng et al. 2021] Derni Ageng, Chin-Ya Huang, Ray-Guang Cheng (2021): ”A short-term household load forecasting framework using lstm and data preparation”; IEEE Access, vol. 9, pp. 167911–167919.

[Aldoseri et al. 2023] Abdulaziz Aldoseri, Khalifa N Al-Khalifa, and Abdel Magid Hamouda (2023): ”Re-Thinking Data Strategy and Integration for Artificial Intelligence: Concepts, Opportunities, and Challenges”; Applied Sciences,vol. 13(12), pp.7082; MDPI.

[Alsufyani et al. 2021] Abdulmajeed Alsufyani, Youseef Alotaibi, Alaa Omran Almagrabi, Saleh Ahmed Alghamdi, Nawal Alsufyani (2021): ”Optimized intelligent data management framework for a cyber-physical system for computational applications”; Complex and Intelligent Systems, pp. 1–13, Springer.

[Amin et al. 2022] Amin, Farhan and Abbasi, Rashid and Mateen, Abdul and Ali Abid, Muhammad and Khan, Salabat (2022): ” A step toward next-generation advancements in the internet of things technologies”; Sensors, vol. 22 ,no.20, pp. 8072, MDPI.

[Atiquzzaman et al. 2020] Mohammed Atiquzzaman, Neil Yen, Zheng Xu (2020): ”Big data analytics for cyber-physical system in smart city: BDCPS 2019, 28-29 December 2019, Shenyang, China”; Springer Nature, vol. 1117.

[Bagchi et al. 2022] Sourav Bagchi, Mamata Jenamani, Aurobinda Routray (2022): ”Multivariate real-time Missing Value Imputation using Adaptive VAR with IoT data from a Refrigerated Container”; in Proceedings of the 2022 IEEE 7th International conference for Convergence in Technology (I2CT), pp. 1–7.

[Caillault et al. 2020] Émilie Poisson Caillault, Alain Lefebvre, André Bigand, et al. (2020): ”Dynamic time warping-based imputation for univariate time series data”; Pattern Recognition Letters, vol. 139, pp. 139–147, Elsevier.

[Chen et al. 2022] Xiaobo Chen, Shurong Liang, Zhihao Zhang, Feng Zhao (2022): ”A Novel Spatiotemporal Data Low-Rank Imputation Approach for Traffic Sensor Network”; IEEE Internet of Things Journal, vol. 9, no. 20, pp. 20122–20135.

[Choi et al. 2021] Chanyoung Choi, Haewoong Jung, Jaehyuk Cho (2021): ”An Ensemble Method for Missing Data of Environmental Sensor Considering Univariate and Multivariate Characteristics”; Sensors, vol. 21, no. 22, pp. 7595.

[Da Silva et al. 2020] Agostinho Da Silva, Andreia Dionísio, Isabel Almeida (2020): ”Enabling cyber-physical systems for Industry 4.0 operations: a service science perspective”; Enabling cyber-physical systems for industry 4.0 operations: a service science perspective, no. 8, pp. 838–846, Blue Eyes Intelligence Engineering and Sciences Publication (BEIESP).

[Deepak Adhikari et al. 2021] Deepak Adhikari, Wei Jiang, Jinyu Zhan (2021): ”Imputation using information fusion technique for sensor generated incomplete data with high missing gap”; Microprocessors and Microsystems, pp. 103636.

[Deng et al. 2022] Grace Deng, Cuize Han, David S Matteson (2022): ”Extended missing data imputation via GANs for ranking applications”; Data Mining and Knowledge Discovery, vol. 36, no. 4, pp. 1498–1520, Springer.

[Deng et al. 2022] Yulong Deng, Chong Han, Jian Guo, Linguo Li, Lijuan Sun (2022): ”Online Missing Data Imputation Using Virtual Temporal Neighbor in Wireless Sensor Networks”; Wireless Communications and Mobile Computing, vol. 2022, Hindawi.

[Djenouri et al. 2022] Youcef Djenouri, Asma Belhadi, Djamel Djenouri, Gautam Srivastava, Jerry Chun-Wei Lin (2022): ”Intelligent Deep Fusion Network for Anomaly Identification in Maritime Transportation Systems”; IEEE Transactions on Intelligent Transportation Systems, IEEE.

[Dong et al.2023] Dong, Shaofeng and Zhan, Jinsong and Hu, Wei and Mohajer, Amin and Bavaghar, Maryam and Mirzaei, A(2023): ”Energy-efficient hierarchical resource allocation in uplink-downlink decoupled NOMA HetNets”; IEEE Transactions on Network and Service Management, vol. 20, no. 3, pp. 3380-3395, Sept. 2023, doi: 10.1109/TNSM.2023.3239417.

[Doove et al. 2014] Lisa L Doove, Stef Van Buuren, Elise Dusseldorp (2014): "Recursive partitioning for missing data imputation in the presence of interaction effects"; Computational statistics and data analysis, vol. 72, pp. 92–104, Elsevier.

[Du et al. 2019] Jinghan Du, Haiyan Chen, Weining Zhang (2019): "A deep learning method for data recovery in sensor networks using effective spatio-temporal correlation data"; Sensor Review, vol. 39, no. 2, pp. 208–217.

[Elsanhoury et al. 2022] Farahsari, Elsanhoury, Mahmoud and Mäkelä, Petteri and Koljonen, Janne and Välisuo, Petri and Shamsuzzoha, Ahm and Mantere, Timo and Elmusrati, Mohammed and Kuusniemi, Heidi (2022): " Precision positioning for smart logistics using ultra-wideband technology-based indoor navigation: A review"; IEEE Access,vol. 10 ,pp. 44413–44445,IEEE.

[Flammini 2021] Francesco Flammini (2021): "Digital twins as run-time predictive models for the resilience of cyber-physical systems: a conceptual framework"; Philosophical Transactions of the Royal Society A, vol. 379, no. 2207, pp. 20200369, The Royal Society Publishing.

[França et al. 2021] Cinthya M França, Rodrigo S Couto, Pedro B Velloso (2021): "Missing data imputation in Internet of Things gateways"; Information, vol. 12, no. 10, pp. 425.

[Gao et al. 2015] Zhipeng Gao, Weijing Cheng, Xuesong Qiu, Luoming Meng (2015): "A missing sensor data estimation algorithm based on temporal and spatial correlation"; International Journal of Distributed Sensor Networks.

[Gharghabi et al. 2020] Shaghayegh Gharghabi, Shima Imani, Anthony Bagnall, Amirali Darvishzadeh, Eamonn Keogh (2020): "An ultra-fast time series distance measure to allow data mining in more complex real-world deployments"; Data Mining and Knowledge Discovery, vol. 34, pp. 1104–1135, Springer.

[González-Vidal et al. 2020] Aurora González-Vidal, Punit Rathore, Aravinda S Rao, José Mendoza-Bernal, Marimuthu Palaniswami, Antonio F Skarmeta-Gómez (2020): "Missing data imputation with Bayesian maximum entropy for internet of things applications"; IEEE Internet of Things Journal, vol. 8, no. 21, pp. 16108–16120.

[Hallaji et al. 2021] Ehsan Hallaji, Roozbeh Razavi-Far, Mehrdad Saif (2021): "DLIN: Deep ladder imputation network"; IEEE Transactions on Cybernetics, vol. 52, no. 9, pp. 8629–8641.

[Huamin et al. 2020] TAO Huamin, DENG Qiuqun, XIAO Shanzhu (2020): "Reconstruction of time series with missing value using 2D representation-based denoising autoencoder"; Journal of Systems Engineering and Electronics, vol. 31, no. 6, pp. 1087–1096.

[Huang et al. 2020] Chuanchao Huang, Yu-Wei Chan, Neil Yen (2020): "Data processing techniques and applications for cyber-physical systems (DPTA 2019)"; Springer.

[Jeong et al. 2023] Soohwan Jeong, Chonghyo Joo, Jongkoo Lim, Hyungtae Cho, Sungsu Lim, and Junghwan Kim (2023): "A novel graph-based missing values imputation method for industrial lubricant data"; Computers in Industry, 150, 103937; Elsevier.

[Jiang et al. 2021] Xiao Jiang, Zean Tian, Kenli Li (2021): "A graph-based approach for missing sensor data imputation"; IEEE Sensors Journal, vol. 21, no. 20, pp. 23133–23144.

[Kim et al. 2023] Jiwon Kim, Younghoon Kwak, Sun-Hye Mun, and Jung-Ho Huh (2023): "Imputation of missing values in residential building monitored data: Energy consumption, behavior, and environment information"; Building and Environment, 245, 110919; Elsevier.

[Kök and Özdemir 2020] İbrahim Kök, Suat Özdemir (2020): "DeepMDP: A novel deep-learning-based missing data prediction protocol for IoT"; IEEE Internet of Things Journal, vol. 8, no. 1, pp. 232–243.

[Kuppannagari et al. 2021] Sanmukh R Kuppannagari, Yao Fu, Chung Ming Chueng, Viktor K Prasanna (2021): "Spatio-temporal missing data imputation for smart power grids"; Proceedings of the Twelfth ACM International Conference on Future Energy Systems, pp. 458–465.

[Lee et al. 2021] Farahsari, Elsanhoury, Mahmoud and Mäkelä, Petteri and Koljonen, Janne and Vä Lee, Gyeong Ho and Han, Jaeseob and Choi, Jun Kyun (2021): " MPdist-based missing data imputation for supporting big data analyses in IoT-based applications Future Generation Computer Systems,vol. 125 ,pp. 421–432, Elsevier.

[Liu et al. 2020] Yuehua Liu, Tharam Dillon, Wenjin Yu, Wenny Rahayu, Fahed Mostafa (2020): "Missing value imputation for industrial IoT sensor data with large gaps"; IEEE Internet of Things Journal, vol. 7, no. 8, pp. 6855–6867.

[Majidi et al. 2021] Babak Majidi, Omid Hemmati, Faezeh Baniardalan, Hamid Farahmand, Alireza Hajitabar, Shahab Sharafi, Khadije Aghajani, Amir Esmaeili, Mohammad Taghi Manzuri (2021): "Geo-spatiotemporal intelligence for smart agricultural and environmental eco-cyber-physical systems"; Enabling AI applications in data science, pp. 471–491, Springer.

[Mary and Arockiam 2017] I Priya Stella Mary, L Arockiam (2017): "Imputing the missing values in IoT using ESTCP model"; International Journal of Advanced Research in Computer Science, vol. 8, no. 9.

[Mohajer et al.2022] Mohajer, Amin and Sorouri, F and Mirzaei, A and Ziaeddini, A and Rad, K Jalali and Bavaghar, Maryam(2022): "Energy-aware hierarchical resource management and backhaul traffic optimization in heterogeneous cellular networks"; EEE Systems Journal, vol. 16,no. 4,pp.5188–5199.

[Mohajer et al. 2023] Mohajer, Amin and Daliri, Mahya Sam and Mirzaei, A and Ziaeddini, A and Nabipour, M and Bavaghar, Maryam(2022) :" Heterogeneous computational resource allocation for NOMA: Toward green mobile edge-computing systems"; IEEE Transactions on Services Computing, vol.16, no. 2, pp. 1225–1238.

[Mohammed et al. 2023] Mohammed, Bzhar Ghafour and Hasan, Dler Salih (2023):"Smart Healthcare Monitoring System Using IoT"; International Journal of Interactive Mobile Technologies ,vol. 17,no. 1,pp. 141–152.

[Muralidhar et al. 2019] Nikhil Muralidhar, Sathappan Muthiah, Kiyoshi Nakayama, Ratnesh Sharma, Naren Ramakrishnan (2019): "Multivariate long-term state forecasting in cyber-physical systems: A sequence to sequence approach"; in Proceedings of the 2019 IEEE International Conference on Big Data (Big Data), pp. 543—552.

[Naik et al. 2023] Kevin Naik and Supriya Patel (2023): "An open source smart home management system based on IoT"; Wireless Networks, 29(3), 989–995; Springer.

[Ni and Cao 2022] Qingjian Ni, Xuehan Cao (2022): "MBGAN: An improved generative adversarial network with multi-head self-attention and bidirectional RNN for time series imputation"; Engineering Applications of Artificial Intelligence, vol. 115, Elsevier.

[Nower et al. 2013] Naushin Nower, Yasuo Tan, Azman Osman Lim (2013): "Efficient spatial data recovery scheme for cyber-physical system"; in Proceedings of the 2013 IEEE 1st International Conference on Cyber-Physical Systems, Networks, and Applications (CPSNA), pp. 72–77.

[Nower et al. 2014] Naushin Nower, Yasuo Tan, Azman Osman Lim (2014): "Efficient temporal and spatial data recovery scheme for stochastic and incomplete feedback data of cyber-physical systems"; in Proceedings of the 2014 IEEE 8th International Symposium on Service Oriented System Engineering, pp. 192–197.

[Okafor and Delaney 2021] Nwamaka U Okafor, Declan T Delaney (2021): "Missing data imputation on IoT sensor networks: Implications for on-site sensor calibration"; IEEE Sensors Journal, vol. 21, no. 20, pp. 22833–22845.

[Razavi-Far et al. 2021] [Razavi-Far et al. 2021]Roozbeh Razavi-Far, Ehsan Hallaji, Maryam Farajzadeh-Zanjani, Ranim Aljoudi, Mehrdad Saif (2021): "A Critical Study on the Impact of Missing Data Imputation for Classifying Intrusions in Cyber-Physical Water Systems"; in Proceedings of IECON 2021–47th Annual Conference of the IEEE Industrial Electronics Society, pp. 1–6.

[Schreiber et al. 2023] Jonas Fernando Schreiber, Airam Sausen, Mauricio De Campos, Paulo Sérgio Sausen, and Marco Thomé Da Silva Ferreira Filho (2023): ”Data Imputation Techniques Applied to the Smart Grids Environment”; IEEE Access, 11, 31931–31940; IEEE.

[Song and Szafir 2018] Hayeong Song, Danielle Albers Szafir (2018): ”Where’s my data? evaluating visualizations with missing data”; IEEE transactions on visualization and computer graphics, vol. 25, no. 1, pp. 914–924, IEEE.

[Syu et al. 2023] Jia-Hao Syu, Gautam Srivastava, Marcin Fojcik, Rafał Cupek, Jerry Chun-Wei Lin (2023): ”Energy grid management system with anomaly detection and Q-learning decision modules”; Computers and Electrical Engineering, vol. 107, article 108639, Elsevier.

[Turabieh et al. 2019] Hamza Turabieh, Majdi Mafarja, Seyedali Mirjalili (2019): ”Dynamic adaptive network-based fuzzy inference system (D-ANFIS) for the imputation of missing data for internet of medical things applications”; IEEE Internet of Things Journal, vol. 6, no. 6, pp. 9316—9325, IEEE.

[Villalonga et al. 2020] Alberto Villalonga, Gerardo Beruvides, Fernando Castano, Rodolfo E Haber (2020): ”Cloud-based industrial cyber–physical system for data-driven reasoning: A review and use case on an industry 4.0 pilot line”; IEEE Transactions on Industrial Informatics, vol. 16, no. 9, pp. 5975–5984, IEEE.

[Wu et al. 2022] Xuesong Wu, Mengyun Xu, Jie Fang, Xiongwei Wu (2022): ”A multi-attention tensor completion network for spatiotemporal traffic data imputation”; IEEE Internet of Things Journal, vol. 9, no. 20, pp. 20203–20213.

[Yan et al. 2015] Xiaobo Yan, Weiqing Xiong, Liang Hu, Feng Wang, Kuo Zhao (2015): ”Missing value imputation based on Gaussian mixture model for the Internet of Things”; Mathematical Problems in Engineering, vol. 2015, Hindawi.

[Yang et al. 2022] Yihong Yang, Xuan Yang, Mohsen Heidari, Mohammad AYOUB Khan, Gautam Srivastava, Mohammad Khosravi, Lianyong Qi (2022): ”Astream: Data-stream-driven scalable anomaly detection with accuracy guarantee in IIoT environment”; IEEE Transactions on Network Science and Engineering, IEEE.

[Yoon et al. 2018] Jinsung Yoon, James Jordon, Mihaela Schaar (2018): ”Gain: Missing data imputation using generative adversarial nets”; International conference on machine learning, pp. 5689–5698, PMLR.

[Zaid et al. 2021] Yemeni Zaid, Bo Zhang, Waleed M Ismael, Yingjuan Xie, Given Name Surname, Haibin Wang (2021): ”ST-MLR: A Spatio-temporal Multiple Linear Regression Missing Data Reconstruction Approach for Improving WSN Data Reliability”; in Proceedings of the 2021 International Congress of Advanced Technology and Engineering (ICOTEN), pp. 1–6.

[Zhao et al. 2016] Liang Zhao, Zhikui Chen, Zhennan Yang, Yueming Hu, Mohammad S Obaidat (2016): ”Local similarity imputation based on fast clustering for incomplete data in cyber-physical systems”; IEEE Systems Journal, vol. 12, no. 2, pp. 1610–1620.

[Zhang et al. 2020] Wenjie Zhang, Yonghong Luo, Ying Zhang, Dipti Srinivasan (2020): ”SolarGAN: Multivariate solar data imputation using generative adversarial network”; IEEE Transactions on Sustainable Energy, vol. 12, no. 1, pp. 743–746, IEEE.

[Zou et al. 2020] Zou, Ning and Liang, Shaobo and He, Daqing (2020): ” Issues and challenges of user and data interaction in healthcare-related IoT: a systematic review”;vol. 38,no. 4,pp. 873—884, Emerald Publishing Limited.