INDIAN JOURNAL OF SCIENCE AND TECHNOLOGY

RESEARCH ARTICLE

OPEN ACCESS

Received: 09-01-2023

Accepted: 30-03-2023

Published: 21-04-2023

A Novel Two Tier Missing at Random Type Missing Data Imputation using Enhanced Linear Interpolation Technique on Internet of Medical Things

P Iris Punitha[1]*, **J G R Sathiaseelan**[2]

1 Assistant Professor, Department of Computer Applications, Bishop Heber College (Affiliated to Bharathidasan University), Tiruchirappalli-620017, Tamil Nadu, India
2 Associate Professor, Department of Computer Science, Bishop Heber College (Affiliated to Bharathidasan University), Tiruchirappalli-620017, Tamil Nadu, India

Citation: Iris Punitha P, Sathiaseelan JGR (2023) A Novel Two Tier Missing at Random Type Missing Data Imputation using Enhanced Linear Interpolation Technique on Internet of Medical Things. Indian Journal of Science and Technology 16(16): 1192-1204. https://doi.org/10.17485/IJST/v16i16.60

* **Corresponding author**.

irispunitha.ca@bhc.edu.in

Funding: None

Competing Interests: None

Published By Indian Society for Education and Environment (iSee)

ISSN
Print: 0974-6846
Electronic: 0974-5645

Abstract

Objectives: Data collection and distribution are essential components required for the victory of Internet of Medical Things (IoMT) system. Generally, missing data is the most recurrent problem that impacts an overall system performance. **Methods:** Missing data in IoMT systems can be caused by various factors, including faulty connections, external attacks, or sensing errors. Although missing data is ubiquitous in IoT, missing data imputation is hardly ever observed in an IoMT setting. As a result, doing analytics on IoMT data with missing values causes a deterioration in the accuracy and dependability of the data analysis outputs. To achieve excellent performance, missing data must be imputed once it occurs in such systems. Therefore, this paper proposes a novel Two Tier Missing Data Imputation (TT-MDI) technique for missing at random (MAR) type missing data in IoMT using an enhanced linear interpolation technique. **Findings**: The proposed TT-MDI algorithm has two tiers for imputing MAR missing data and it was tested using the Kaggle Machine Learning Repository's cStick IoMT dataset. Utilizing the distances between the class centroids with their related data instances, the first tier aims to identify the imputation threshold. The identified threshold is then used by the second tier to impute missing data. According to the experimental findings, the proposed work offers higher accuracy, precision, recall, and F-measure for imputed dataset using the TT-MDI technique than missing data included dataset when compared to the original dataset. **Novelty:** The TT-MDI technique consists of two tiers. The first tier uses Manhattan distances between class centroids and related data instances to discover the imputation threshold. Next, the second tier uses the discovered threshold to impute missing data using the Enhanced Linear Interpolation technique.

Keywords: Internet of Medical Things; Imputation of Missing Data; Threshold Discovery; Manhattan Distance

1 Introduction

The Internet of Medical Things (IoMT) is a system of hardware elements, software and medical devices connected to the Internet for use in healthcare IoT[1]. IoMT, also known as the Internet of Things in healthcare, enables distant and wireless devices to safely interact via the Internet to enable quick and appropriate healthcare data processing.

The Internet of Things has a clear and long-lasting impact on the healthcare industry. According to a recent Deloitte survey[2], the worldwide IoMT market is expected to grow from $41 billion in 2017 to $158 billion by 2022. The IoMT has a variety of effects on the healthcare industry. These changes are most noticeable when IoMT is utilized in the home, on the body, in the community, and hospitals.

1.1 In-Home IoMT

Utilizing in-home-IoMT, individuals can transmit healthcare records from their residences to other locations, like their healthcare practitioner or a medical Centre. For instance, remote patient monitoring (RPM) involves using medical devices to transmit data from recently discharged patients' hospitals, such as pulse rate and oxygen saturation, for evaluation by their doctors. Through the early detection of issues, re-admissions to hospitals may be avoided.

1.2 On-Body IoMT

Worn medical devices linked to remote surveillance or tracking systems are on-body-IoMT. On-body-IoMT, unlike in-home-IoMT, could be utilized exterior of the house as individual go about their daily activities.

Consumer-on-body-IoMT devices are worn health trackers which anybody could purchase for personal use or share with physicians. These devices can provide early warning indicators for further severe health issues and record a typical metric like heart rate. The Apple Watch, for instance, could alert users to abnormal cardiac beats.

1.3 Community IoMT

The utilization of IoMT sensors across a larger city or geographical region is known as community-IoMT. Community-IoMT includes technology that allows remote services and mobile and emergency treatment. Point-of-care devices, for instance, could be utilized through health professionals in non-conventional hospital contexts like military hospitals, and kiosks could be utilized to distribute medicines to patients in locations where conventional infrastructures are limited or non-existent.

1.4 In-Hospital IoMT

Hospitals need to monitor the availability and distribution of their medical resources throughout time and how staff and patients shift about the facility. Healthcare practitioners use IoMT devices and other surveillance equipment to monitor these communications; thus, those administrators can obtain a complete picture of what is going on.

The most serious problem with IoT devices is a shortage of crucial data required to keep them working. Missing data is a serious issue that hinders the effectiveness of programs and forces them to fail[3]. A robust technique for retrieving missing data should ensure that IoMT systems work properly and precisely[4].

For example, suppose a problem (e.g., a sensor failure) happens in the motion sensor. In that scenario, the data from the sensor won't be adequately received by the desired applications, which would cause significant issues for users (e.g., doctors) who will miss vital information about the patient's status. We provide a unique technique for filling missing data utilizing a novel Two Tier Missing Data Imputation (TT-MDI) technique that considers the previous data obtained by a given sensor to fore see novel plausible missed values in IoMT systems to minimize potential difficulties.

(cStick[5]), a well-known medical dataset which models IoMT systems is used to validate the proposed technique. The effectiveness of the IoMT systems will suffer if there are missing values in this dataset (especially missing at random (MAR) type). To maintain IoMT networks' functionality and guarantee that customers receive top quality services, missing data must be filled in. Threshold discovery (Tier-1) and Missing data imputation (Tier-2) utilizing Enhanced Linear Interpolation are the two primary tiers of the proposed TT-MDI technique for imputing missing data.

The following sections make up the paper's configuration: In Section II, the related work of existing missing data imputation approaches is discussed. Section III demonstrates the proposed technique for imputing missing data. In Section IV, the experimental setup, as well as the analysis and assessment of the attained outcomes, are described. Section V provides the conclusion at the end.

2 Related Work

This section discussed some related works of existing missing data imputation techniques.

Jia et al.[5] suggested a mixed matrix factorization-based Imputation Method for traffic Congestion Records, or CIM for short. CIM jointly analyzes the regularity, road similarities, and temporal synchronization features of traffic congestion behaviors in order to approximate the missing congestion data. In specifically, using the data on traffic congestion, the authors initially built an order-3 tensor. The authors then used joint matrix factorization to predict the periodicity and road similarity by combining the geographical and temporal data. In order to guarantee the temporal coherence, the authors added local constraints to the matrix factorization process. According to their findings on an actual traffic dataset, modeling all three characteristics of traffic patterns at once is effectual, and CIM outperformed the benchmarks for imputation work for missing traffic records.

Turabieh et al.[6] presented a D-ANFIS (dynamic adaptive network-based fuzzy inference system) approach for missing data imputation but accurately. The main contribution is to impute missing values after they are obtained by splitting the acquired data into two datasets: 1) complete dataset (with no missing data) and 2) incomplete dataset (with missing data) (with missing data). Then, the D-ANFIS is trained using a holdout approach with complete data, and the missing value is imputed using the partial dataset (s). The authors concluded that using D-ANFIS improves IoMT performance effectively.

Silva-Ramirez et al.[7] initiated the CANFIS-ART technique to automate data imputation, which is using the Co-active Neuro-Fuzzy Inference System. The Fuzzy- ART technique is used to create this model, consisting of Neural Network adaptive abilities and a fuzzy logic qualitative method. The efficacy of the CANFIS-ART approach is evaluated to that of various modern imputation methods such as Multilayer Perceptron or Hot-Deck, among others, utilizing a total of eighteen databases showing a perturbation procedure using the random formation of non-monotone missing data patterns. The fields, variables, and sizes of the data sets vary. A set of three classifiers was used to compare databases imputed by various approaches. These findings were statistically examined utilizing Wilcoxon signed-rank test. The authors determined that the CANFIS-ART method surpasses existing approaches and shows a greater generalization power, enhancing the accuracy of the data in databases containing missing data.

Su et al.[8] presented a strategy for imputing missing data using statistics and machine learning that takes advantage of one-dimensional interpolation of the interest variable to capture global trends and linear compensatory of multidimensional variables to capture local variation. First, the mapping of multidimensional nonlinear variables into a feature space utilizing KCPA, and the resulting novel variables are directly coupled by the interest variable. The linear compensation is then performed using these novel variables in conjunction with the multidimensional linear variables. According to the comparative experiment, this technique surpasses commonly utilized techniques by decreasing RMSE by 29.19 per cent and MAE by 44.73 per cent on average and having the nearby to 1.

França et al.[9] provided a technique to forecast and impute missing data in IoT gateways to attain special autonomy on the network edge. These gateways normally have inadequate computing resources. As a result, missing data imputation approaches must be simple and effective. As a result, two regression methods using neural networks were given in this paper to impute incomplete data in IoT gateways. The authors looked at the execution time and the quantity of memory required, and the forecast quality. The authors used six years of climate data from Rio de Janeiro to evaluate their models, altering the missing data percentages. The authors concluded that neural network regression methods work better than the other imputation approaches tested, based on averages and repetition of past values for overall missing data percentages. Furthermore, the neural network models have a short execution time and use fewer than 140 KB of memory, allowing them to operate on IoT gateways quickly.

Kamkhad et al.[10] proposed and explored a novel method for semantically imputing missing data using an ontology model. The authors add three novel developments to the area: Initially, they use Particle Swarm Optimization (PSO) that is applied to the problem of cleaning up integer data, to increase the effectiveness of missing data forecasting. PSO's effectiveness is enhanced utilizing K-means to assist determine the fitness score. Next, using ontology with PSO to reduce the search space and improve PSO's accuracy in anticipating missing numerical values whilst also allowing it to arrive at a solution swiftly. The last step is to create a logical framework for structural and knowledge-based modelling that will use conceptual linkages to substitute the nominal data that is lacking from the dataset. As evaluated through the root-mean-square error, the authors determined that their methodology can calculate missing data effectively and with less likelihood of error than traditional techniques.

Lee et al.[11] propose MP-BMDI, a high-performance imputation algorithm for sustaining big data studies in IoT systems, where the lack of huge missing sub-sequences is necessary to provide unbiased outcomes. The authors use a method that involves finding a finite amount of sub-sequences which are generally identical to the sub-sequence before the missing data, after that adjusting the tallness of the sub-sequences that follow to ideal places. The relevant sub-sequence completely replaces the missing gap after the most appropriate sub-sequence for replacement is picked between them using the pattern score function PSF(r) proposed in[12]. By exploiting sensor data acquired from real environment surveillance and providing substantial insights on the algorithm's efficacy from many viewpoints, mathematical outcomes are offered to confirm the algorithm's advantages compared to alternative benchmark methodologies.

Ji et al.[13] advocated using multi-source data to fill in the missing traffic data. Gru network captures missing patterns because of traffic data's regularity and specificity. The processed missing, mask, and time interval data are fed into the Gru network for additional in-depth information capture. The findings of road speed matching for floating vehicle information on the road in the relevant period are investigated further by the Gru network. The two outcomes are combined to provide the filling value of the missing value.

Utilizing an unsupervised neural network and Adaptive Resonance Theory 2 (ART2), Nickolas et al.[14] introduced CLUSTIMP, a Clustering Based Imputation Algorithm. The effectiveness of the imputation approach is evaluated to existing imputation algorithms using the Root Mean Squared Error (RMSE) rate as the performance indicator. In addition, the impact of the methodology is assessed utilizing imputation data set for classification accuracy, with a Type II error rate decrease ranging from 2% to 11% depending on the classifier.

3 Methodology

3.1 Novel Two Tier Missing At Random Type Missing Data Imputation

This section proposed a novel two-tier missing data imputation technique (TT-MDI) to impute the missing at the random type of data. Utilizing the distances between the class centroids with their related data instances, the first tier aims to identify the imputation threshold. The identified threshold is then used by the second tier to impute missing data. The subsections that follow describe these two tiers.

3.2 Tier 1 - Threshold Discovery

Figure 1 depicts the first tier's process for identifying the threshold for missing value imputation. It consists of four steps, each of which is explained below.

Step 1: Given dataset D has N classes and M dimensions.

Step 2: The i-th class (i = 1 to N) of D, represented as D_i, is separated into complete ($D_{i\_complete}$) and incomplete subsets ($D_{i\_incomplete}$), where the latter contain missing data.

Figure 2 illustrates nine feature dimensions with a two-class incomplete dataset, in which the red boxes specify the missing values.

Step 3: For the i-th class, compute its class centroid of j-th feature, cent(D_{ij}), which is explained in Equation (1), and standard deviation std(D_{ij}) of $D_{i\_complete,}$ which is explained in Equation (2).

$$\text{cent}\left(\mathrm{D_{ij}}\right) = \frac{\sum x_{ij}}{n} \tag{1}$$

Here $\sum x_{ij}$ is the sum of j-th feature values from all i-th class data instances, and n is the number of i-th class data instances.

$$\text{std}\left(\mathrm{D_{ij}}\right) = \sqrt{\frac{\sum \left(x_{ij} - \text{cent}(\mathrm{Dij})\right)^2}{n}} \tag{2}$$

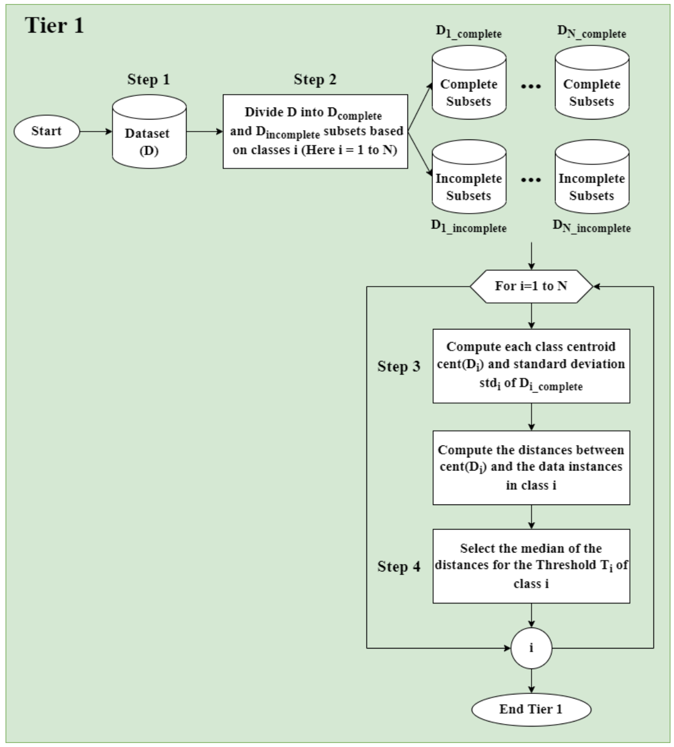

Fig 1. Flow diagram of Tier1-Threshold Discovery

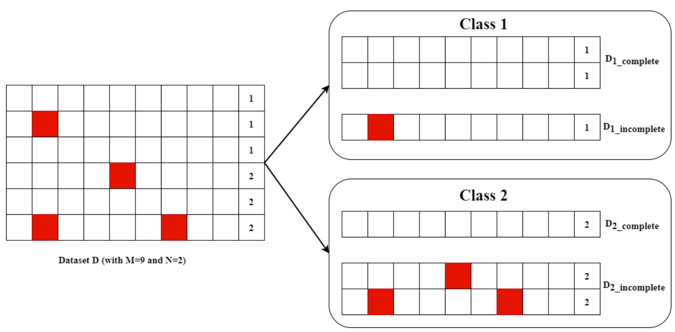

Fig 2. A two-class dataset example for subset division

Using Equation (1) and Equation (2), we can compute the class centroid and standard deviation of all features and obtains cent(D_i) and std(D_i). Next, the Manhattan distance (MD) between cent(D_i) and each data instance in class i is computed. For example, let the values of a cent(D_i) is [a_1,…., a_m] and the values of a data instance is [b_1,…., b_m], then the Manhattan distance calculation shown in Equation (3).

$$MD = |a_1 - b_1| + \ldots + |a_m - b_m| \tag{3}$$

Step 4: The threshold Ti for class i is determined by taking the median of the distances between each data instance in the class and the class centroid. Finally, Steps 3 and 4 are repeated up until the threshold is attained for each class.

Figure 3 demonstrates an illustration of computing the 1st class centroid cent(D_1), the standard deviation std(D_1), the distances between cent(D_1) with other data instances and threshold value computation using the median.

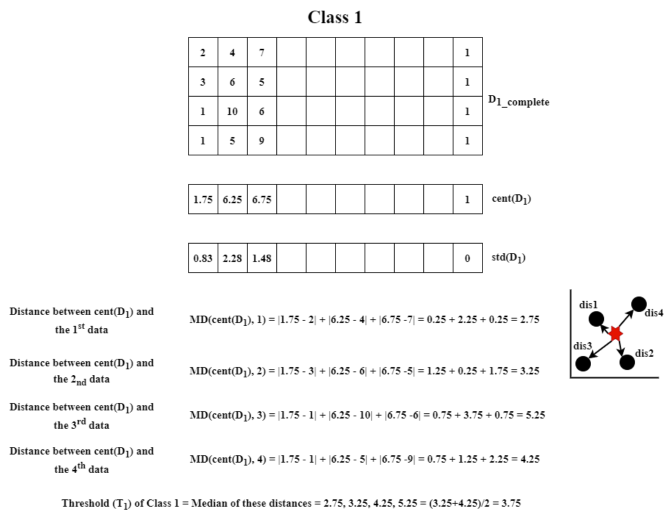

Fig 3. An illustration of class1 for the class centroid, standard deviation, Manhattan distances and median

The Threshold discovery algorithm showed in Algorithm 1.

Algorithm 1: Threshold discovery		
Input	:	Dataset D with M feature dimensions, N classes, and P data instances
Output	:	N threshold values for N classes
Step 1	:	For j = 1 to P
Step 2	:	If D(j) has missing value(s) then
Step 3	:	Get the class label of D(j) and set this class label to the variable i
Step 4	:	Put D(j) to $D_{i\_incomplete}$
Step 5	:	Else
Step 6	:	Get the class label of D(j) and set this class label to the variable i
Step 7	:	Put D(j) to $D_{i\_complete}$
Step 8	:	End If
Step 9	:	End For
Step 10	:	For i = 1 to N
Step 11	:	cent($D_{i\_complete}$) = Mean($D_{i\_complete}$) // **Eq. (1)**
Step 12	:	std($D_{i\_complete}$) = Standard Deviation($D_{i\_complete}$) // **Eq. (2)**
Step 13	:	End For
Step 14	:	For j = 1 to P
Step 15	:	MD_$D_{i_complete}$(j) = Manhattan distance(cent($D_{i\_complete}$), $D_{i\_complete}$(j)) // **Eq. (3)**
Step 16	:	End For
Step 17	:	For i = 1 to N
Step 18	:	Threshold(i) = Median(MD_$D_{i_complete}$)
Step 19	:	End For

3.3 Tier 2 – Missing Data Imputation

Figure 4 shows the missing data imputation flow for Tier 2. It includes the three steps outlined below.

Step 1: For the i-th class incomplete dataset ($D_{i\_incomplete}$) contains Num_miss_data instances: If the data instance has only one missing value, proceed to Step 2; otherwise, if it has several missing values, proceed to Step 3.

Step 2: For $D_{i\_incomplete}$, when the data j has only one missing value, the class centre cent($D_{i\_complete}$) is utilized to impute the missing value of j. Figure 5 illustrates an illustration for the 1st class incomplete dataset, i.e., $D_{1\_incomplete}$. In this illustration, as the third feature of data j is missing, the third feature of a cent($D_{1\_complete}$) is substituted for the missing data.

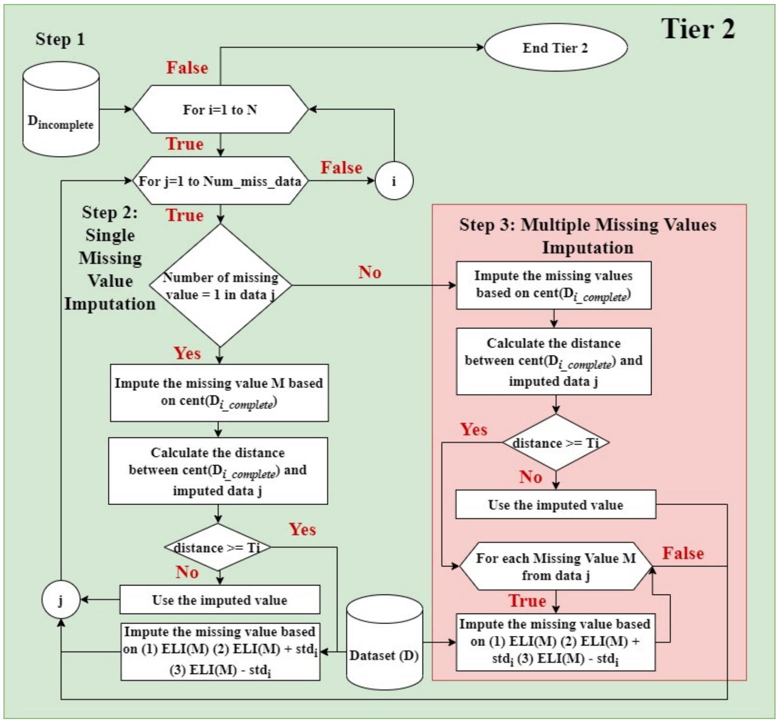

Fig 4. Flow diagram of Tier 2 - Missing Data Imputation

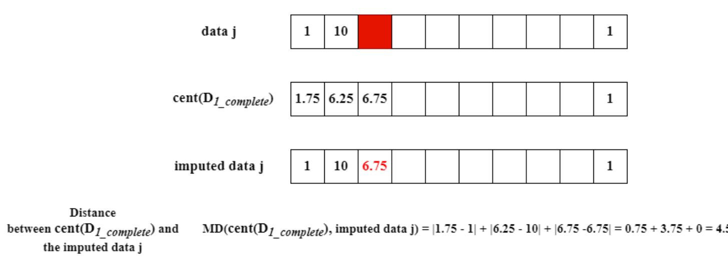

Fig 5. An example of class centroid based missing value imputation

Subsequently, the distance between the imputed data j and cent($D_{1\_complete}$) is computed to compare it with the threshold T_i. The imputation procedure for data j is done if the distance is smaller than Ti.

Otherwise, three various values are imputed if the distance is greater than or equal to Ti. These three values are computed using Linear Interpolation (LI) and standard deviation. A curve fitting method called linear interpolation generates novel data points within the bounds of a finite collection of known data points using linear polynomials. Two neighboring identified values measured before and after X are required for linear interpolation. Suppose the value at point X is missing. In that case, the value at point X will be computed using both the final actual assessment performed before point X, identifying point A, and the first actual assessment taken after point X, identifying point B. (a value), (b value), and (x value) are the values at positions A, B, and X, respectively. At X observation, Equation (4) computes the missing x value.

$$LI\,(x\ value) = \frac{1}{2}(a\ value + b\ value) \tag{4}$$

Figure 6 also shows an example of missing data imputation using linear interpolation. In addition, linear interpolation is rapid and simple; however, it is not accurate in a dataset with class labels. As a result, an Enhanced Linear Interpolation (ELI) method is proposed to improve linear interpolation accuracy. This method considers initial and subsequent values, previous class values, current class values, and subsequent class values. For example, suppose the value at point X is missing. In that case, the value at point X will be computed using both the last actual assessment performed before point X, identifying point A, and the first actual assessment taken after point X, identifying point B. (a value, a class), (b value, b class), and (x value, x class) are the coordinates at positions A, B, and X, correspondingly. For X observation, Equation (5) estimates the missing value x value at x class.

$$ELI\,(x\ value) = \frac{1}{2}\left(LI\,(x\ value) + \left[b\ value\ + \left[\frac{(a\ value - b\ value) * (x\ class - b\ class)}{(a\ class - b\ class)} \right] \right] \right) \tag{5}$$

The example of enhanced linear interpolation based on missing data imputation is also explained in Figure 6. The enhanced linear interpolation is based on imputed data taken as imputed result 1. After enhanced linear interpolation based on missing data imputation, the imputed value is added by standard deviation, taken as imputed result two. Finally, the imputed value is subtracted by standard deviation, taken as imputed result 3.

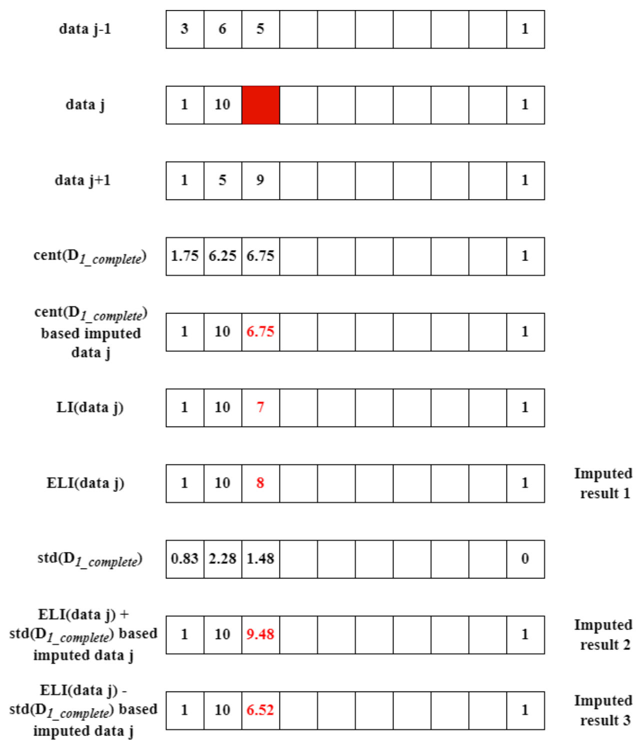

Fig 6. An example of class centroid, standard deviation, linear interpolation and enhanced linear interpolation based on missing data imputation.

After that, the distance between all imputed results (for data j) and cent(Di_complete) is computed in Figure 7. Then, the imputed result with a smaller distance than the other imputed results is taken as the final imputed result.

Distance between cent(D_1) and the imputed result 1	MD(cent($D_{1\_complete}$), imputed result 1) = |1.75 - 1| + |6.25 - 10| + |6.75 -8| = 0.75 + 3.75 + 1.25 = 5.75
Distance between cent(D_1) and the imputed result 2	MD(cent($D_{1\_complete}$), imputed result 2) = |1.75 - 1| + |6.25 - 10| + |6.75 -9.48| = 0.75 + 3.75 + 2.73 = 7.23
Distance between cent(D_1) and the imputed result 3	MD(cent($D_{1\_complete}$), imputed result 3) = |1.75 - 1| + |6.25 - 10| + |6.75 -6.52| = 0.75 + 3.75 + 0.23 = 4.73

Fig 7. Distance between all imputed results with a cluster centroid.

In Figure 7, imputed result 3 has a smaller distance than the others. So we can take imputed result three as the final imputed result**.**

Step 3: The appropriate feature values of a cent(Di) are imputed for any numerous missing values of data j. Next, a comparison using the threshold Ti is made to determine the distance between the imputed data j and cent(Di). An illustration of imputing two missing values for class 1 is shown in Figure 8.

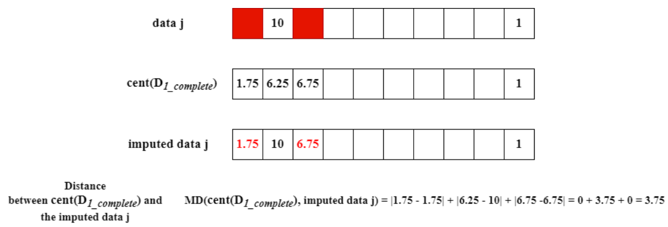

Fig 8. A case where two missing values were imputed

As a result, the imputation process for data j is finished if the distance is less than T_i. Otherwise, each missing value is successively imputed by ELI(data j), adding and subtracting the std_i with ELI(data j) of the imputed attribute values, if appropriate, if the distance is larger than or equal to T_i.

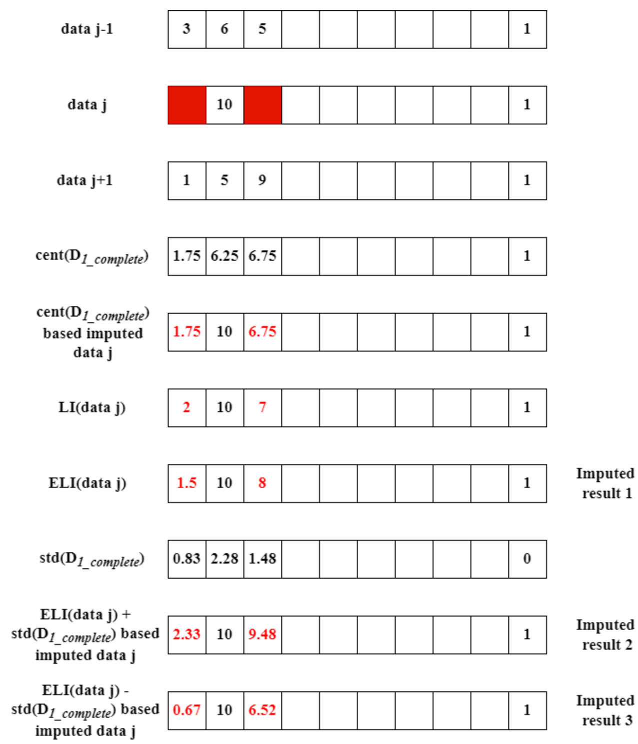

Fig 9. A case where two missing values were imputed

After that, the distance between all imputed results (for data j) and cent($D_{i\_complete}$) is computed, which is explained in Figure 10. Then, the imputed result with a smaller distance than the other imputed results is taken as the final imputed result.

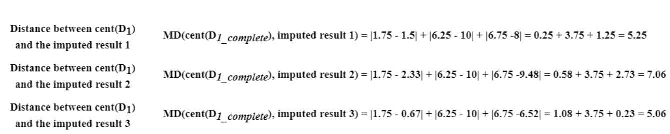

Fig 10. Distance between all imputed results with a cluster centroid

In Figure 10, imputed result 3 has a smaller distance than the others. Therefore, the imputed result 3 is considered the final imputed result. The proposed Missing data imputation process (Tier-2) showed in Algorithm 2.

Algorithm 2: Missing Data Imputation		
Input	:	$D_{i\_incomplete}$ containing N classes and M feature dimensions
Output	:	Imputed dataset for $D_{i\_incomplete}$
Step 1	:	For i = 1 to N
Step 2	:	For j=1 to Num_miss_data **// Num_miss_data - Number of available missing data instances in ith class**
Step 3	:	If the number of missing values = 1 in data j
Step 4	:	Impute the missing value M based on cent($D_{i\_complete}$)
Step 5	:	Distance = Manhattan distance(cent($D_{i\_complete}$), $D_{i\_incomplete}$(j))
Step 6	:	If Distance > Threshold(i)
Step 7	:	Impute the missing value based on (1) ELI(M) (2) ELI(M) + std_i (3) ELI(M) - std_i
Step 8	:	Else
Step 9	:	Use the imputed value
Step 10	:	End If
Step 11	:	Else
Step 12	:	Impute the missing values based on cent($D_{i\_complete}$)
Step 13	:	Distance = Manhattan distance(cent($D_{i\_complete}$), $D_{i\_incomplete}$(j))
Step 14	:	If Distance > Threshold(i)
Step 15	:	For each Missing Value M in data j
Step 16	:	Impute the missing value based on (1) ELI(M) (2) ELI(M) + std_i (3) ELI(M) - std_i
Step 17	:	End For
Step 18	:	Else
Step 19	:	Use the imputed value
Step 20	:	End If
Step 21	:	End If
Step 22	:	End For
Step 23	:	End For

4 Results and Discussion

This study uses the "cStick dataset" obtained from the Kaggle machine learning repository[5]. There are 2039 instances and seven features in all. For evaluation, we randomly inserted 10% missing data in the original dataset (data of 203 instances missing according to MAR type). The TT-MDI technique is used to recover missing data. Threshold discovery and missing data imputation are the two tiers of this approach. According to the results, before imputation and after imputation the proposed work offers higher accuracy for imputed dataset using the TT-MDI technique than MAR missing data included dataset when compared to the original dataset. The proposed TT-MDI technique was evaluated utilizing accuracy, precision, recall, and f-measure using Machine Learning-based classifiers, namely the Repeated Incremental Pruning to Produce Error Reduction (RIPPER), Support Vector Machine (SVM), Naive Bayes (NB) and C4.5 classifiers. The classifiers are implemented using the Weka tool. The Experimental results clearly shows that all evaluation metrics were enhanced by 4 % to 14 % after imputing missing data using the TT-MDI algorithm.

4.1 Accuracy

The definition of accuracy is the ratio of the total number of correct forecasts to the total number of forecasts. The per cent of each accurately predicted data point was denoted by accuracy, as illustrated in Equation (6).

$$\text{Accuracy} = \frac{(\mathrm{TP}+\mathrm{TN})}{(\mathrm{TP}+\mathrm{TN}+\mathrm{FP}+\mathrm{FN})} \tag{6}$$

Table 1 demonstrates the attained findings before and after missing data imputation based on the Accuracy value using the TT-MDI algorithm.

Table 1. Accuracy comparison of before and after missing data imputation using the TT-MDI technique

Data	Accuracy (in %)			
	RIPPER	SVM	NB	C4.5
cStick Original dataset	97.5728	91.4572	86.0346	91.1451
cStick Missing dataset	79.6865	79.9896	74.5061	77.6084
cStick Imputed Dataset	93.315	83.2034	81.6316	90.017

Table 1 compares accuracy before and after employing the TT-MDI algorithm to fill in missing data. The results show that utilizing the TT-MDI technique to impute missing value(s) improved the RIPPER, SVM, NB and C4.5 classifiers final performance. The accuracy of the TT-MDI technique, for example, is improved by almost 12% once missing data are imputed. While minor, this change aids the IoMT applications in performing well and continuing to function without missing data issues.

4.2 Precision

The ratio of True Positives to all Positives is known as precision. Precision measures exactness or quality and a positive predictive value, as illustrated in Equation (7).

$$Precision = \frac{TP}{(TP+FP)} \tag{7}$$

Table 2 demonstrates the findings attained before and after missing data was imputed based on Precision value using the TT-MDI algorithm.

Table 2. Precision comparison of before and after missing data imputation using the TT-MDI algorithm

Data	Precision (in %)			
	RIPPER	SVM	NB	C4.5
cStick Original dataset	96.473	90.8308	87.7065	92.385
cStick Missing dataset	81.7431	74.7614	69.8278	80.0692
cStick Imputed Dataset	91.0901	87.435	79.8693	84.9774

Table 2 compares precision before and after employing the TT-MDI algorithm to fill in missing data. The results show that utilizing the TT-MDI technique to impute missing value(s) improved the RIPPER, SVM, NB and C4.5 classifiers final performance. The precision value of the TT-MDI technique, for example, is increased by approximately 9% once missing data are imputed. This change, while minor, aids the IoMT applications in performing well and continuing to function without missing data issues.

4.3 Recall

Recall is the measure of accurate identification of true positives. The recall is also known as sensitivity shown in Equation (8).

$$Recall = \frac{TP}{(TP+FN)} \tag{8}$$

The attained findings before and after missing data imputation based on the Recall value using the TT-MDI algorithm is shown in Table 3.

Table 3. Recall comparison of before imputing missing data and after imputing missing data using the TT-MDI algorithm

Data	Recall (in %)			
	RIPPER	SVM	NB	C4.5
cStick Original dataset	96.2309	88.6585	87.3535	94.5988
cStick Missing dataset	77.1882	72.3559	67.8606	79.4142
cStick Imputed Dataset	87.0738	86.4643	79.1077	87.2304

Table 3 compares recall before and after employing the TT-MDI technique to fill in missing data. The results show that utilizing the TT-MDI technique to impute missing value(s) improved the RIPPER, SVM, NB and C4.5 classifiers final performance. The recall value of the TT-MDI technique, for example, is increased by approximately 11% once missing data are imputed. This change, while minor, aids the IoMT applications in performing well and continuing to function without missing data issues. It is clear from Table 3 that the overall recall after imputing missing data using the TT-MDI algorithm was enhanced.

4.4 F-Measure

The harmonic mean of recall and precision is called the F-measure, shown in Equation (9).

$$F - Measure = 2.\frac{(Precision * Recall)}{(Precision + Recall)} \tag{9}$$

The attained findings before and after missing data imputation based on the F-measure value using the TT-MDI algorithm is shown in Table 4.

Table 4. F-measure comparison of before and after missing data imputation using the TT-MDI algorithm

Data	F-Measure (in %)			
	RIPPER	**SVM**	**NB**	**C4.5**
cStick Original dataset	96.3518	89.7315	87.5296	93.4788
cStick Missing dataset	79.4004	73.539	68.8301	79.7403
cStick Imputed Dataset	89.0367	86.9469	79.4866	86.0892

Table 4 compares F-measure before and after using the TT-MDI technique to impute missing data. The results show that utilizing the TT-MDI technique to impute missing value(s) improved the RIPPER, SVM, NB and C4.5 classifiers final performance. After missing data is imputed, the F-measure value utilizing the TT-MDI technique increases by approximately 10%. This change, while minor, aids the IoMT applications in performing well and continuing to function without missing data issues. It is clear from Table 4 that the all F-measure after missing data imputation using the TT-MDI algorithm is enhanced.

5 Conclusion

Data with a missing value poses a concern to IoMT systems because it is the most frequent problem that degrades system performance in general. As a result, end-user satisfaction declines. Data loss in IoMT networks can be caused by several factors, including faulty connections, external attacks, and sensor failures. Missing value handling is a difficult yet fascinating field of research in data mining and information retrieval. Missing data must be imputed as soon as it happens in these systems to guarantee excellent performance. Therefore, this paper proposed a novel Two Tier Missing Data Imputation (TT-MDI) algorithm for MAR type missing data in the IoMT. Utilizing the distances between the class centroids with their related data instances, the first tier aims to identify the imputation threshold. The identified threshold is then used by the second tier to impute missing data. The proposed work offers higher accuracy for imputed dataset using the TT-MDI technique than missing data included dataset when compared to the original dataset. This technique was evaluated using accuracy, precision, recall, and f-measure using Machine Learning-based classifiers, namely the Repeated Incremental Pruning to Produce Error Reduction (RIPPER), Support Vector Machine (SVM), Naive Bayes (NB) and C4.5 classifiers. Based on the accuracy, precision, recall, and f-measure, the obtained results proved which the TT-MDI algorithm is advantageous in increasing the entire efficiency of IoMT applications.

References

1) Al-Turjman F, Nawaz MH, Ulusar UD. Intelligence in the Internet of Medical Things era: A systematic review of current and future trends. *Computer Communications*. 2020;150:644–660. Available from: https://doi.org/10.1016/j.comcom.2019.12.030.
2) Deloitte M. How connected medical devices are transforming health care. 2018. Available from: https://www2.deloitte.com/content/dam/Deloitte/global/Documents/Life-Sciences-Health-Care/gx-lshc-medtech-iomt-brochure.pdf.
3) Manickam P, Mariappan SA, Murugesan SM, Hansda S, Kaushik A, Shinde R, et al. Artificial Intelligence (AI) and Internet of Medical Things (IoMT) Assisted Biomedical Systems for Intelligent Healthcare. *Biosensors*. 2022;12(8):562–562. Available from: https://doi.org/10.3390/bios12080562.
4) Yildirim E, Cicioğlu M, Çalhan A. Real-time internet of medical things framework for early detection of Covid-19. *Neural Comput & Applic*. 2022;34:20365–20378. Available from: https://doi.org/10.1007/s00521-022-07582-x.
5) Jia X, Dong X, Chen M, Yu X. Missing data imputation for traffic congestion data based on joint matrix factorization. *Knowledge-Based Systems*. 2021;225:107114–107114. Available from: https://doi.org/10.1016/j.knosys.2021.107114.
6) Turabieh H, Mafarja M, Mirjalili S. 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*. 2019;6(6):9316–9325. Available from: https://doi.org/10.1109/JIOT.2019.2926321.

7) Silva-Ramirez EL, Cabrera-Sánchez JFF. Correction to: Co-active neuro-fuzzy inference system model as single imputation approach for non-monotone pattern of missing data. *Neural Computing and Applications*. 2022;34:2495–2496. Available from: https://doi.org/10.1007/s00521-021-06623-1.
8) Su T, Shi Y, Yu J, Yue C, Zhou F. Nonlinear compensation algorithm for multidimensional temporal data: A missing value imputation for the power grid applications. *Knowledge-Based Systems*. 2021;215:106743–106743. Available from: https://doi.org/10.1016/j.knosys.2021.106743.
9) França CM, Couto RS, Velloso PB. Missing Data Imputation in Internet of Things Gateways. *Information*. 2021;12(10):425–425. Available from: https://doi.org/10.3390/info12100425.
10) Kamkhad N, Jampachaisri K, Siriyasatien P, Kesorn K. Toward semantic data imputation for a dengue dataset. *Knowledge-Based Systems*. 2020;196:105803–105803. Available from: https://doi.org/10.1016/j.knosys.2020.105803.
11) Ji Z, Zhu W. Dynamic L-RNN recovery of missing data in IoMT applications. 2018. Available from: https://doi.org/10.1016/j.future.2018.07.006.
12) Hasan MK, Alam MA, Roy S, Dutta A, Jawad MT, Das S. Missing value imputation affects the performance of machine learning: A review and analysis of the literature. *Informatics in Medicine Unlocked*. 2021;27:100799–100799. Available from: https://doi.org/10.1016/j.imu.2021.100799.
13) Ji Z, Zhu W. A traffic data imputing method based on multisource recurrent neural network. *International Conference on Computer Application and Information Security (ICCAIS 2021)*. 2022;12260:90–95. Available from: https://doi.org/10.1117/12.2637835.
14) Nickolas S, Shobha K. Clustering Based Imputation Algorithm Using Unsupervised Neural Network for Enhancing the Quality of Healthcare Data. *Journal of Ambient Intelligence and Humanized Computing*. 2021;12:1771–1781. Available from: https://doi.org/10.1007/s12652-020-02250-1.