

Clustering-based Self-Supervised Multi-Scale Generative Adversarial Network for Data Imputation

Yi Xu¹✉^[0000-0001-6722-5587], Xuhui Xing¹ ^[0009-0009-6109-4817], Anchi Chen¹^[0009-0004-6283-7161],
Yang Liu¹^[0009-0004-7869-1945]

¹ Anhui University, Hefei, 230601, China
xuyi1023@126.com

Abstract. Missing data has always been a challenging issue in machine learning. The Generative Adversarial Imputation Network (GAIN) has been proven to be superior to many existing solutions. However, GAIN suffers from two limitations: first, it does not consider the correlations among input samples; second, it only imputes based on adversarial loss and reconstruction loss of non-missing values without considering the reconstruction loss of missing values. To address these issues, this paper proposes a clustering-based self-supervised multi-scale Generative Adversarial Network for data imputation method, CCGAIN. Firstly, the dataset to be imputed is clustered, and subsequent imputation is performed on samples within each cluster. Then, based on features with low missing rates, local scale data is constructed for each cluster. Next, we use the imputation results of local scale missing values as supervised information for global scale missing value imputation, constructing the reconstruction loss for global scale missing values. Finally, based on the reconstruction loss of missing values, the reconstruction loss of non-missing values, and the adversarial loss, imputation is performed at the global scale. Experimental results demonstrate the effectiveness of this method.

Keywords: Missing Data, Generative Adversarial Networks, Clustering.

1 Introduction

With the rapid advancement and popularization of information technology, the global volume of data is experiencing a sharp increase. Within these data lie numerous valuable pieces of information [1]. However, during the process of data collection and storage, data missing often occurs due to various reasons such as temporary inability to collect or record loss [2]. Such situations can limit a comprehensive understanding of the real scenario, reduce the accuracy and reliability of data analysis and modeling, and even lead to erroneous decision-making [3]. Fig. 1 shows incomplete data with dimension 6. A simple method to handle missing data is to directly delete samples or variables containing missing values. However, when the amount of missing data is substantial, this approach may result in information loss in the dataset, hence nowadays, most research tends towards data imputation. Traditional imputation methods are mainly based on statistical analysis [4], but they make strict assumptions about the data distribution.

If the data does not meet these assumptions, it may affect the accuracy of the imputation results. Furthermore, statistical methods often assume that the relationships between data are linear or simple functional relationships, making it difficult to capture complex patterns and nonlinear relationships in the data, which limits the improvement of imputed data quality. The emergence and rapid development of deep learning now provide new ideas and methods for solving real-world problems [5]. Generative Adversarial Imputation Network (GAIN) [6], as a mainstream missing data imputation model, has been proven to outperform many existing methods.

x_{11}	x	x_{13}	x	x_{15}	x_{16}
x	x_{22}	x	x_{24}	x_{25}	x
x_{31}	x_{32}	x_{33}	x	x_{35}	x_{36}

non-missing value missing value

Fig. 1. Examples of incomplete data with dimension 6.

However, GAIN has two drawbacks: firstly, GAIN does not consider the correlations among input samples. Secondly, during the process of imputing missing values using the generator in GAIN, as there are no ground truth values for missing values to serve as supervised information, it is not possible to construct a reconstruction loss for missing values. Therefore, the rationality of imputed values can only be judged through the reconstruction loss of non-missing values and the adversarial loss. These two limitations restrict the improvement of data imputation quality.

This paper proposes a clustering-based self-supervised multi-scale Generative Adversarial Network data imputation algorithm (CCGAIN) by clustering samples with high correlations together and constructing different scales of data hierarchy. The imputation results of missing values at the local scale are used as supervised information for imputing missing values at the global scale. Firstly, the dataset to be imputed is clustered, grouping samples with high relevance into clusters to effectively utilize the intrinsic relationships among samples. Then, based on features with low missing rates, local scale data is constructed for each cluster. Next, the imputation results of local scale missing values are used as supervised information to construct the reconstruction loss for global scale missing values. Finally, based on the reconstruction loss of missing values, the reconstruction loss of non-missing values, and the adversarial loss, imputation is performed at the global scale. Experimental results demonstrate that the CCGAIN model outperforms mainstream algorithms, especially when the missing rate is high, this superiority is more pronounced. The main contributions of this work are as follows:

- A clustering module is introduced to effectively address the insufficient consideration of inter-sample correlations in GAIN when handling data.
- A multi-scale data construction method is proposed based on features with low missing rates for the GAIN network.

- By utilizing the imputation results of missing values at partial scales as supervised information for imputing missing values at the global scale, the reconstruction loss of missing values is constructed, thereby enhancing the quality of data imputation.
- We demonstrate that our approach surpasses existing methods in both imputation and prediction accuracy, particularly under conditions of high missing data rates.

2 Related Work

2.1 Imputation Methods

Methods for handling missing data can mainly be divided into two categories: deletion and imputation [7, 8]. Deletion [9] involves directly removing samples or features with missing values from the dataset to ensure data integrity and accuracy. This method is suitable for situations where the amount of missing data is small or the missing data has little impact on the analysis results. However, excessive deletion may lead to a reduction in the amount of data, affecting the reliability and effectiveness of the analysis [10]. Therefore, imputation methods are commonly used to handle missing data. Currently, imputation methods can be divided into two main categories: traditional machine learning-based imputation and deep learning-based imputation.

Traditional machine learning-based imputation methods: The K-Nearest Neighbor (KNN) imputation algorithm [11] uses the values of K nearest neighbors to impute missing data. However, since this algorithm needs to traverse the entire dataset for each missing value imputation, the efficiency of the KNN algorithm significantly decreases when dealing with large-scale datasets [12]. The Multiple Imputation by Chained Equations (MICE) algorithm [13] first assigns a random value to each missing value, and then updates the value of the specified variable with the values of other variables in a series of iterations until the algorithm converges to complete the imputation. However, for a large number of missing data, the MICE algorithm is less efficient, time-consuming, and based on simple models such as linear regression or logistic regression for prediction, which may not capture complex relationships in the data. The MissForest algorithm [14] treats variables with missing data as labels and other variables as features. It trains a random forest model using the training set consisting of labels without missing data and their corresponding feature data, and then uses this model to predict missing values. However, the MissForest algorithm is based on random forest for prediction, and such complex models may lead to overfitting, resulting in inaccurate imputation results. The Expectation Maximization (EM) algorithm [15] first estimates the parameter values of the model based on existing data and then predicts missing data based on these parameter values. These two steps are iteratively repeated until convergence to impute data. However, since the EM algorithm is an iterative process, it may get stuck in local optima and fail to reach the global optimum.

Deep learning-based imputation methods: In recent years, due to the powerful non-linear fitting capability of neural networks, deep generative models have been used to impute missing data. Gondara and Wang [16] proposed a multiple imputation model DAE based on deep denoising autoencoders to impute data. Nazabal A. Olmos, Ghahramani, and others [17] proposed a general framework for variational autoencoders that

can effectively fill incomplete heterogeneous data. Spinelli et al. [18] designed missing data imputation based on graph denoising autoencoders, where each edge of the graph encodes the similarity between two patterns. Lai et al. [19] proposed an architecture called Tracking Removal Autoencoder (TRAE), which dynamically redesigns the input structure of hidden neurons based on traditional autoencoders. Nazabal et al. [20] proposed a comprehensive framework called HI-VAE for constructing variational autoencoders tailored to fitting incomplete heterogeneous data. Richardson et al. [21] proposed MCFlow, which leverages normalizing flow generative models and Monte Carlo sampling for imputation. With the continuous development of deep learning technology, Generative Adversarial Networks (GANs) have become the latest research hotspot. Recently, GANs have also been applied in the field of data imputation [22, 23]. Awan et al. [24] proposed a model CGAIN, which performs missing data imputation by utilizing class-specific distributions to generate optimal estimates for the missing values. Yoon and Sull [25] proposed GAMIN which divides into unconditional and conditional generators. It aims to improve upon GAIN's performance with high-dimensional missing data. Qiu et al. [26] proposed a feature-specific deep adversarial imputation pipeline capable of accurately imputing various types of input data. Li and Jiang [27] proposed MisGAN to learn and impute images from complex high-dimensional incomplete data, introducing an auxiliary GAN to learn the mask distribution to simulate missing situations. Zhang et al. [28] proposed CPM-GAN, which aims to learn a unified latent representation by considering both completeness and structure, to impute missing data by utilizing the correlation between different modalities. Wang et al. [29] proposed PCGAIN, which learns latent class information from a low missing rate data subset and trains an auxiliary classifier based on synthetic pseudo-labels, integrating this classifier into the GAN to help the generator generate higher quality imputation results. Yoon et al. [6] proposed the Generative Adversarial Imputation Network (GAIN), which optimizes the generator using the adversarial loss of the GAN and the reconstruction loss of non-missing values to improve the imputation quality of the generator.

Although these methods have achieved certain effects in imputing missing data, due to the lack of sufficient supervised information and inadequate consideration of the correlation between input samples, these imputation methods still have various deficiencies in learning the distribution of original data, resulting in low imputation accuracy.

2.2 GAIN

In this subsection, we introduce the traditional GAIN network. The architecture of the traditional network is shown in Fig. 2. In the imputation process of the model, it can be divided into two parts: the generator and the discriminator.

Assuming the original dataset $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_N\} \in \mathbb{R}^{N \times d}$, among them, the i -th data vector $\mathbf{x}_i = \{x_{i1}, x_{i2}, \dots, x_{id}\} \in \mathbb{R}^{1 \times d}$, x_{ij} represents the j -th feature component of the i -th data vector in \mathbf{X} . Given dataset $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_N\} \in \mathbb{R}^{N \times d}$, $\mathbf{x}_i = \{x_{i1}, x_{i2}, \dots, x_{id}\} \in \mathbb{R}^{1 \times d}$,

- (1) For each $\mathbf{x}_i \in \mathbf{X}$, define the corresponding mask vector $\mathbf{m}_i = \{m_{i1}, m_{i2}, \dots, m_{id}\} \in \{0,1\}^d$, when x_{ij} is not missing, $m_{ij} = 1$, and when x_{ij} is missing, $m_{ij} = 0$. All mask vectors constitute the mask matrix $\mathbf{M} = \{\mathbf{m}_1, \mathbf{m}_2, \dots, \mathbf{m}_N\}$.
- (2) For each $\mathbf{x}_i \in \mathbf{X}$, define a new random vector $\mathbf{z}_i = \{z_{i1}, z_{i2}, \dots, z_{id}\} \in \mathbb{R}^{1 \times d}$, all random vectors \mathbf{z}_i form a random matrix $\mathbf{Z} = \{\mathbf{z}_1, \mathbf{z}_2, \dots, \mathbf{z}_N\}$.
- (3) For each $\mathbf{x}_i \in \mathbf{X}$, define a new data vector $\tilde{\mathbf{x}}_i = \{\tilde{x}_{i1}, \tilde{x}_{i2}, \dots, \tilde{x}_{id}\} \in \mathbb{R}^{N \times d}$, the elements within it, $\tilde{x}_{ij} = \begin{cases} x_{ij}, & \text{if } m_{ij} = 1 \\ 0, & \text{if } m_{ij} = 0 \end{cases}$. Here, the missing value * is replaced with 0. All data vectors form the data matrix $\tilde{\mathbf{X}} = \{\tilde{\mathbf{x}}_1, \tilde{\mathbf{x}}_2, \dots, \tilde{\mathbf{x}}_N\}$.

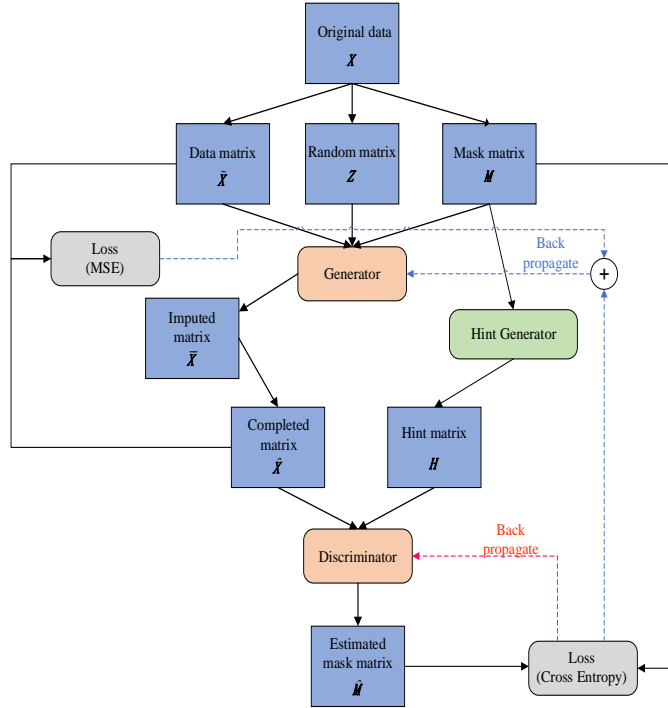


Fig. 2. The architecture of GAIN

The generator of GAIN takes three parts as input: the data matrix $\tilde{\mathbf{X}}$, the random matrix \mathbf{Z} , and the mask matrix \mathbf{M} . The generator G of GAIN generates data at all positions in the matrix, regardless of whether it is in the location of missing value or non-missing value. It learns the data distribution of the input data and then outputs an imputed matrix $\bar{\mathbf{X}}$, the same size as the data matrix $\tilde{\mathbf{X}}$, with a distribution that approximates the real data, the process is:

$$\bar{\mathbf{X}} = G(\tilde{\mathbf{X}}, \mathbf{M}, (1 - \mathbf{M}) \odot \mathbf{Z}) \quad (1)$$

where \odot represents the Hadamard product. The imputed matrix $\bar{\mathbf{X}}$ is a new matrix generated by the generator G to mimic the data distribution of the data matrix. Obviously, $\bar{\mathbf{X}}$ is not the desired imputation result, because there are some values in the data matrix $\tilde{\mathbf{X}}$ that are not missing, and not all positions require imputation. By concatenating the data matrix $\tilde{\mathbf{X}}$ with the imputed matrix $\bar{\mathbf{X}}$, replacing the non-missing data in $\tilde{\mathbf{X}}$ with the corresponding data at the same position in $\bar{\mathbf{X}}$, a newly completed matrix $\hat{\mathbf{X}}$ is formed, which is the final imputation result. The specific process is as follows:

$$\hat{\mathbf{X}} = \mathbf{M} \odot \mathbf{X} + (1 - \mathbf{M}) \odot \bar{\mathbf{X}} \quad (2)$$

The loss function of generator G is defined as:

$$\min_G \frac{1}{N} \sum_{k=1}^N (L_G(\mathbf{m}_k, \hat{\mathbf{m}}_k) + \alpha L_R(\tilde{\mathbf{x}}_k, \bar{\mathbf{x}}_k)) \quad (3)$$

$$L_G(\mathbf{m}_i, \hat{\mathbf{m}}_i) = \sum_{j=1}^d (-(1 - m_{ij}) \log \hat{m}_{ij}) \quad (4)$$

$$L_R(\tilde{\mathbf{x}}_i, \bar{\mathbf{x}}_i) = \sum_{j=1}^d m_{ij} (\tilde{x}_{ij} - \bar{x}_{ij})^2 \quad (5)$$

where α is a weight parameter, $\hat{\mathbf{m}}_k$ is the estimated mask vector output by the discriminator D . L_G is part of the adversarial loss of GAIN, which plays a role in competing with the discriminator D . By backpropagation, the generator G is optimized to make the discriminator D unable to distinguish between imputed data and non-missing data. L_R is the reconstruction loss of non-missing value. The purpose of L_R is to minimize the distance between the values generated by generator G at the positions of non-missing data and the non-missing data, i.e., to require generator G to learn the distribution of non-missing data, which is the distribution of real data.

The discriminator of GAIN takes as input the hint matrix \mathbf{H} and the completed matrix $\hat{\mathbf{X}}$. The hint matrix \mathbf{H} is generated by feeding the mask matrix \mathbf{M} into the hint generator, which provides some information about \mathbf{M} to the discriminator D . When discriminating between real and fake data, GAIN evaluates all components of the completed matrix $\hat{\mathbf{X}}$. In GAIN, the discriminator D outputs a probability for each position in the matrix, forming an estimated mask matrix $\hat{\mathbf{M}}$. The component \hat{m}_{ij} of $\hat{\mathbf{M}}$ represents the probability that D judges the component \hat{x}_{ij} of the completed matrix $\hat{\mathbf{X}}$ as non-missing data. The entire input-output process can be represented as follows:

$$\hat{\mathbf{M}} = D(\hat{\mathbf{X}}, \mathbf{H}) \quad (6)$$

the mask matrix \mathbf{M} is the ground truth of the estimated mask matrix $\hat{\mathbf{M}}$, meaning that the closer the output values of discriminator D are to the values of the mask matrix \mathbf{M} , the better the discriminative ability of discriminator D .

The definition of the loss function of discriminator D is as follows:

$$\min_D \frac{1}{N} \sum_{k=1}^N L_D(\mathbf{m}_k, \hat{\mathbf{m}}_k) \quad (7)$$

$$L_D(\mathbf{m}_i, \hat{\mathbf{m}}_i) = \sum_{j=1}^d (-m_{ij} \log \hat{m}_{ij} - (1 - m_{ij}) \log(1 - \hat{m}_{ij})) \quad (8)$$

L_D is also part of the adversarial loss, which plays a role in competing with the generator G . It aims to maximize the ability of discriminator D to distinguish between non-missing data and imputed data in the input data.

Although GAIN performs well in data imputation, it still has two limitations:

(1) GAIN fails to consider the correlation between input samples, leading to low imputation accuracy. In data imputation, the model's input samples should be closely related to the data to be imputed to effectively predict missing values. Taking the KNN (K-Nearest Neighbors) imputation algorithm [30] as an example, this algorithm first calculates the K nearest neighbors in the dataset with the highest correlation to the data to be imputed. After determining the neighbors, their feature values can be used for weighted averaging or voting prediction to fill in missing values. The advantage of this method lies in its consideration of the similarity between data, resulting in more accurate and reliable imputed data. However, GAIN lacks this consideration.

(2) GAIN relies solely on adversarial loss and reconstruction loss of non-missing values for imputation, without considering the reconstruction loss of missing values. In the imputation process of GAIN, the positions of components in the data matrix can be divided into two parts based on whether the data is missing at that position. The positions with missing data are referred to as missing value positions, while the positions where data is retained are referred to as non-missing value positions. Although GAIN's generator generates data at both types of positions, it mainly focuses on the generation of data at non-missing value positions when learning the distribution. This is because when generating data at non-missing value positions, the generator has observed data as a reference, which is real data and retains the distribution of real data, implying that these data can be used as supervision information. However, for data at missing value positions, besides using adversarial theory and letting the discriminator judge whether it is reasonable, GAIN has no other means to ensure the accuracy of imputation. Because the data at missing value positions are already lost and do not have the condition to use their true values as supervision information, it is impossible to construct reconstruction loss for missing values. In this case, only adversarial loss is relied upon to evaluate the rationality of the imputed values, which leads to a decrease in the quality of generated data.

3 CCGAIN

We propose CCGAIN, a clustering-based self-supervised multi-scale generative adversarial network method for data imputation, addressing the two limitations of GAIN. Firstly, to tackle the limitation of GAIN in neglecting the correlation between input samples, CCGAIN conducts clustering on the input data before imputation. Subsequent imputation is performed within each cluster. Secondly, addressing the inability of GAIN to construct reconstruction loss for missing values, CCGAIN initially constructs multi-scale data, then, it uses the imputation results of missing values at the local scale as supervision for missing values at the global scale, constructing the reconstruction loss for missing values. Based on this loss, along with the reconstruction loss for non-missing values and adversarial loss, imputation is performed at the global scale. Based

on the above treatments for the two limitations, CCGAIN consists of two main steps: clustering process and imputation, where imputation involves multi-scale construction and imputation processes. These steps will be discussed separately below.

3.1 Clustering Process

GAIN fails to consider the limitation of inter-sample correlation in its input data. In this regard, CCGAIN addresses this by clustering the input data before imputation. The process of clustering, which divides the dataset into multiple clusters, can provide important guidance for the imputation model, enabling it to conduct data imputation more targeted based on the characteristics of each cluster. Initially, CCGAIN employs GAIN to impute missing data within the dataset, resulting in a complete dataset conducive to subsequent clustering operations. Subsequently, the dataset is partitioned into multiple clusters, with each cluster representing a collection of samples with similar features. Upon imputation of the partitioning, the imputed data within each cluster is reverted to its missing state, preserving the original missing data structure. Finally, each cluster's data is individually taken as the input dataset for subsequent imputation operations, and then all clusters are merged to form the final imputation result. This targeted approach leverages the high correlation within clusters, thereby enhancing the quality of imputation results in CCGAIN. Fig. 3 shows the flowchart of clustering processing for CCGAIN.

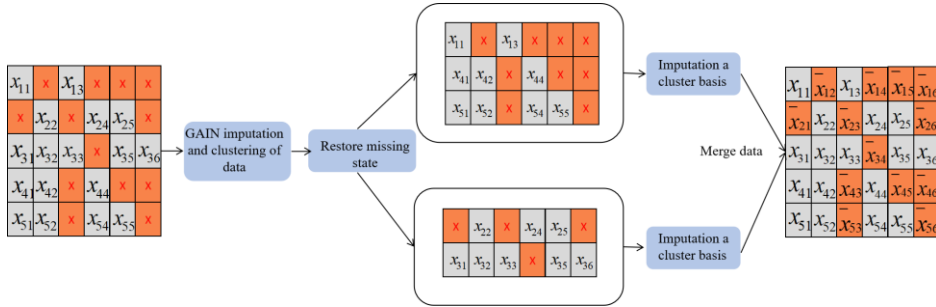


Fig. 3. Flowchart of clustering processing for CCGAIN

3.2 Multi-scale Construction Method

In this subsection, we introduce the multi-scale construction method of CCGAIN, which forms the basis for the subsequent imputation process. The goal is to construct local scale data for each cluster. All data within the cluster serve as the global scale data, this means that the imputation results at the global scale represent the imputation results for the entire set of samples.

For a given dataset $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_N\} \in \mathbb{R}^{N \times d}$, Assuming F represents the set of all features in the dataset, $F = \{f_1, f_2, \dots, f_d\}$, Where f_i represents the i -th feature in the dataset, $1 \leq i \leq d$. Firstly, provide the definition formula for the missing rate of features:

For a given dataset $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_N\} \in \mathbb{R}^{N \times d}$, $\mathbf{x}_i = \{x_{i1}, x_{i2}, \dots, x_{id}\} \in \mathbb{R}^{1 \times d}$, $\mathbf{m}_i = \{m_{i1}, m_{i2}, \dots, m_{id}\}$ is the mask vector of \mathbf{x}_i , $F = \{f_1, f_2, \dots, f_d\}$ represents the set of all features in the dataset. For feature f_j , $1 \leq j \leq d$, the missing rate $r(f_j)$ is defined as follows:

$$r(f_j) = \frac{1}{N} \sum_{i=1}^N m_{ij} \quad (9)$$

The features are arranged in ascending order of their missing rates, and the top $d/2$ features are selected. The data of these selected features are used as local scale data. This approach is based on the assumption that data with lower missing rates retain more real data information. Additionally, this way helps to preserve the distribution of the original data as much as possible. Moreover, this procedure supports the subsequent imputation of missing values at the local scale by providing supervision for imputing missing values at the global scale.

The first $d/2$ features of the sorted feature set can be denoted as $F' = \{f'_1, f'_2, \dots, f'_{\frac{d}{2}}\}$. The data described by the local feature F' is represented as $\tilde{\mathbf{X}}' = \{\tilde{\mathbf{x}}'_1, \tilde{\mathbf{x}}'_2, \dots, \tilde{\mathbf{x}}'_N\} \in \mathbb{R}^{N \times \frac{d}{2}}$, $\tilde{\mathbf{x}}'_i = \{\tilde{x}'_{i1}, \tilde{x}'_{i2}, \dots, \tilde{x}'_{i\frac{d}{2}}\} \in \mathbb{R}^{1 \times \frac{d}{2}}$, \tilde{x}'_{ij} represents the j -th feature component of the i -th data vector in $\tilde{\mathbf{X}}'$. All data within the cluster will be treated as data on a global scale, just rearranged according to the ascending order of feature missing rates for convenience in calculating the reconstruction loss of missing values on the global scale.

3.3 Imputation Process

This subsection provides a detailed explanation of the imputation process in CCGAIN. The imputation in CCGAIN is performed on a per-cluster basis, after imputing each cluster using the same method, the clusters are merged to obtain the final imputation result for the entire dataset. For each cluster, imputation begins at the local scale. Missing values are imputed in using GAIN. Then, the imputed values at the local scale serve as supervised information for missing values at the global scale, this process constructs a reconstruction loss for missing values based on the imputation results at the local scale. Utilizing this reconstruction loss, along with the reconstruction loss for non-missing values and an adversarial loss, imputation is performed at the global scale. The imputed results at the global scale for each cluster are then merged to obtain the final imputation result. Fig. 4 shows the input process of CCGAIN.

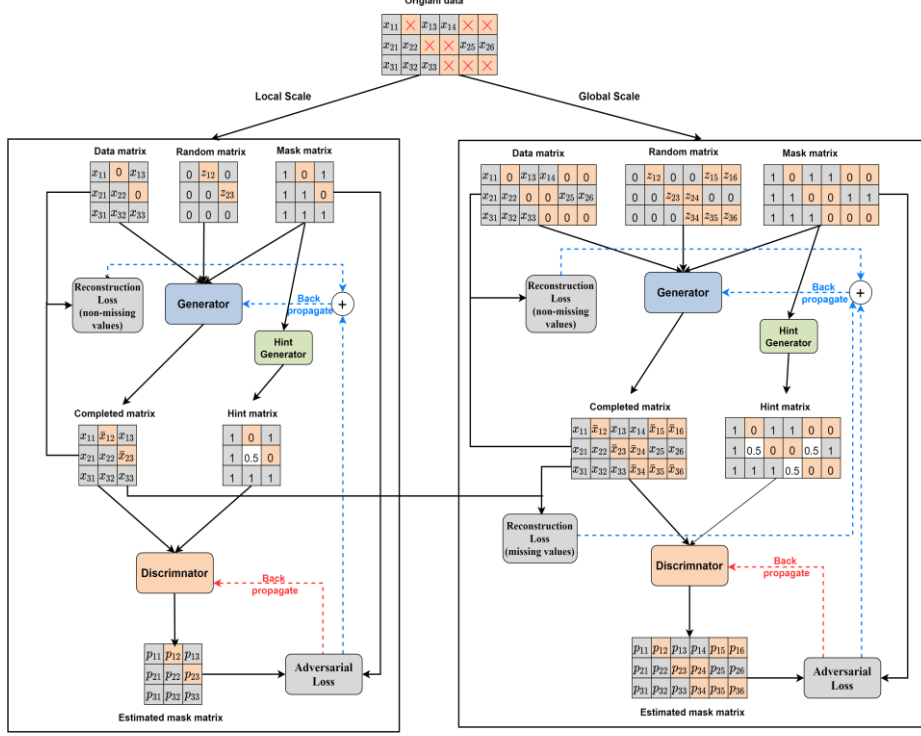


Fig. 4. The imputation process of CCGAIN

(1) Given data $\tilde{\mathbf{X}}' = \{\tilde{\mathbf{x}}'_1, \tilde{\mathbf{x}}'_2, \dots, \tilde{\mathbf{x}}'_N\} \in \mathbb{R}^{N \times \frac{d}{2}}$ at the local scale, for each $\tilde{\mathbf{x}}'_i \in \tilde{\mathbf{X}}'$, there are corresponding binary mask vectors \mathbf{m}'_i and random vectors \mathbf{z}'_i . All mask vectors constitute the mask matrix $\mathbf{M}' = \{\mathbf{m}'_1, \mathbf{m}'_2, \dots, \mathbf{m}'_N\}$. All random vectors form a random matrix $\mathbf{Z}' = \{\mathbf{z}'_1, \mathbf{z}'_2, \dots, \mathbf{z}'_N\}$. Similarly, imputed vector $\bar{\mathbf{x}}'_i$ and completed vector $\hat{\mathbf{x}}'_i$ respectively constitute the imputed matrix $\bar{\mathbf{X}}'$ and the completed matrix $\hat{\mathbf{X}}'$. At the local scale, the traditional GAIN model is used for imputation. Its significance lies in providing supervision for imputation at a global scale. Given the local scale data matrix $\tilde{\mathbf{X}}'$ and its corresponding mask matrix \mathbf{M}' , along with a random matrix \mathbf{Z}' inputted into GAIN, employing the traditional GAIN model for imputation yields the imputed matrix $\bar{\mathbf{X}}'$ and the completed matrix $\hat{\mathbf{X}}'$ for the local scale, this can be represented as:

$$\bar{\mathbf{X}}' = G(\tilde{\mathbf{X}}', \mathbf{M}', (1 - \mathbf{M}') \odot \mathbf{Z}') \quad (10)$$

$$\hat{\mathbf{X}}' = \mathbf{M}' \odot \tilde{\mathbf{X}}' + (1 - \mathbf{M}') \odot \bar{\mathbf{X}}' \quad (11)$$

The input and output of discriminator D are

$$\hat{\mathbf{M}}' = D(\hat{\mathbf{X}}', \mathbf{H}') \quad (12)$$

At this point, the loss function of generator G is

$$\min_G \frac{1}{N} \sum_{k=1}^N (L_G(\mathbf{m}'_k, \hat{\mathbf{m}}'_k) + \alpha L_R(\tilde{\mathbf{x}}'_k, \bar{\mathbf{x}}'_k)) \quad (13)$$

$$L_G(\mathbf{m}'_i, \hat{\mathbf{m}}'_i) = \sum_{j=1}^{d/2} (-(1 - m'_{ij}) \log \hat{m}'_{ij}) \quad (14)$$

$$L_R(\tilde{\mathbf{x}}'_i, \bar{\mathbf{x}}'_i) = \sum_{j=1}^{d/2} m'_{ij} (\tilde{x}'_{ij} - \bar{x}'_{ij})^2 \quad (15)$$

The loss function of discriminator D is

$$\min_D \frac{1}{N} \sum_{k=1}^N L_D(\mathbf{m}'_k, \hat{\mathbf{m}}'_k) \quad (16)$$

$$L_D(\mathbf{m}'_i, \hat{\mathbf{m}}'_i) = \sum_{j=1}^{d/2} (-m'_{ij} \log \hat{m}'_{ij} - (1 - m'_{ij}) \log(1 - \hat{m}'_{ij})) \quad (17)$$

Next, based on their respective loss functions, calculate the losses and utilize stochastic gradient descent to iteratively update the network structures of the generator and discriminator until they reach equilibrium. Finally, obtain the completed matrix $\hat{\mathbf{X}}'$ generated by the generator at the local scale.

(2) At the global scale, assuming the completed matrix obtained at the local scale is $\hat{\mathbf{X}}'$, the calculation method for the imputed matrix $\bar{\mathbf{X}}$ and the completed matrix $\hat{\mathbf{X}}$ remains unchanged. The data matrix $\tilde{\mathbf{X}}$ and its corresponding binary mask matrix \mathbf{M} , along with a random matrix \mathbf{Z} , are inputted into the generator G to obtain the imputed matrix $\bar{\mathbf{X}}$ and the completed matrix $\hat{\mathbf{X}}$ at the global scale.

$$\bar{\mathbf{X}} = G(\tilde{\mathbf{X}}, \mathbf{M}, (1 - \mathbf{M}) \odot \mathbf{Z}) \quad (18)$$

$$\hat{\mathbf{X}} = \mathbf{M} \odot \tilde{\mathbf{X}} + (1 - \mathbf{M}) \odot \bar{\mathbf{X}} \quad (19)$$

The discriminator D takes the completed matrix $\hat{\mathbf{X}}$ and a hint matrix \mathbf{H} as inputs and outputs estimated mask matrix $\hat{\mathbf{M}}$, which predicts which data in $\hat{\mathbf{X}}$ are imputed and which are non-missing.

$$\hat{\mathbf{M}} = D(\hat{\mathbf{X}}, \mathbf{H}) \quad (20)$$

The loss function of discriminator D is

$$\min_D \frac{1}{N} \sum_{k=1}^N L_D(\mathbf{m}_k, \hat{\mathbf{m}}_k) \quad (21)$$

At this stage, the loss function of the generator G has changed. Apart from incorporating the reconstruction loss of non-missing data and the adversarial loss against the discriminator, the loss function of G now utilizes the completed matrix $\hat{\mathbf{X}}'$ at the local scale as the supervision information for generating the completed matrix $\hat{\mathbf{X}}$ at the global scale. This construction leads to the reconstruction loss L_M for missing values at the global scale.

$$L_M(\hat{\mathbf{x}}'_i, \hat{\mathbf{x}}_i) = \sum_{j=1}^{d/2} (\hat{x}'_{ij} - \hat{x}_{ij})^2 \quad (22)$$

The significance of L_M is to use imputed values of missing data at a local scale to supervise the imputation of missing values at a global scale on the first $d/2$ features. For the missing values on the other $d/2$ features, due to the higher missing rate of features, local-scale supervision is not feasible, but the generator uniformly performs imputation on missing values, ensuring that under L_M , the imputed values on the first $d/2$ features fit closer to the original data, while also guaranteeing better accuracy on the imputed values of the remaining $d/2$ features.

The loss function of generator G is

$$\min_G \frac{1}{N} \sum_{k=1}^N (L_G(\mathbf{m}_k, \hat{\mathbf{m}}_k) + \alpha L_R(\tilde{\mathbf{x}}_k, \bar{\mathbf{x}}_k) + \beta L_M(\hat{\mathbf{x}}'_k, \hat{\mathbf{x}}_k)) \quad (23)$$

where α and β are weight parameters.

Next, according to their respective loss functions, calculate the losses and use stochastic gradient descent to iteratively update the network structures of the generator and discriminator until they reach equilibrium. Finally, obtain the completed matrix $\hat{\mathbf{X}}$ at the global scale. The completed matrix at the global scale for each cluster is then merged to obtain the final imputation result.

4 Experiment

4.1 Dataset and experimental details

(1) Dataset. To validate the effectiveness of the proposed CCGAIN model, we conducted experiments using six datasets obtained from the UCI Machine Learning Repository. The specific information about the datasets is presented in Table 1 below.

Table 1. The basic properties of the UCI datasets.

Dataset	Samples	Numerical variables	Categorical variables	Number of classes
Breast Cancer	569	30	0	2
Divorce	170	0	54	2
Letter	20000	16	0	26
News	39797	35	25	2
Sales	811	106	0	0
Valley	606	100	0	2

(2) Experimental methods. baseline methods compared with CCGAIN include EM [15], MissForest [14], KNN [11], MICE [13], MCFlow [21], GAIN [6], DAE [16], and PCGAIN [29]. MissForest [14], MICE [13], and EM [15] belong to machine learning-based imputation models, while DAE [16], MCFlow [21], GAIN [6], and PCGAIN [29] are deep generative methods. Three sets of experiments were conducted. In the first set of experiments, under the condition where missing data accounts for 50% of the total data, CCGAIN was compared with the aforementioned baseline methods on

six datasets for imputation results, with the evaluation metric being Root Mean Square Error (RMSE). Then CCGAIN was compared with GAIN on the six datasets under different percentages of missing data, and RMSE was again used as the evaluation metric for imputation performance. In the second set of experiments, the prediction performance under various missing rates of imputed results by CCGAIN and GAIN was compared. The data was imputed first, and then the class labels of samples were predicted, with the post-imputation prediction accuracy being compared between the two methods. In the third set of experiments, ablation experiments were conducted.

(3) Experimental Parameter Settings. The performance of CCGAIN is relatively stable concerning the number of clusters. In most cases, regardless of the true number of classes, CCGAIN can achieve good results when the value of K is between 2 and 4. Therefore, in practice, only a small number of clusters are needed to save computational costs. Here, we set the value of K uniformly to 4. The clustering method chosen is K-Means.

Table 2. Imputation performance of CCGAIN and comparison methods in terms of RMSE(Average \pm Std of RMSE).

Algorithm	Breast Cancer	Divorce	Letter	News	Sales	Valley
KNN	0.1476 ± 0.0158	0.3753 ± 0.035	0.172 ± 0.014	0.2623 ± 0.023	0.255 ± 0.0257	0.1305 ± 0.0167
EM	0.2020 ± 0.0074	0.3612 ± 0.026	0.215 ± 0.010	0.2736 ± 0.034	0.2926 ± 0.0151	0.1204 ± 0.0342
MICE	0.1129 ± 0.0246	0.3453 ± 0.015	0.1714 ± 0.022	0.2523 ± 0.035	0.2197 ± 0.0157	0.1267 ± 0.0342
MissForest	0.1185 ± 0.0154	0.356 ± 0.036	0.1541 ± 0.017	0.2626 ± 0.038	0.2467 ± 0.0110	0.1321 ± 0.0324
DAE	0.1203 ± 0.0053	0.3862 ± 0.028	0.1628 ± 0.006	0.2478 ± 0.026	0.2634 ± 0.0125	0.1445 ± 0.0103
GAIN	0.085 ± 0.0045	0.3309 ± 0.02	0.1554 ± 0.007	0.2361 ± 0.022	0.2406 ± 0.0190	0.1022 ± 0.0095
PCGAIN	0.1098 ± 0.0059	0.3688 ± 0.022	0.1559 ± 0.004	0.2353 ± 0.023	0.2227 ± 0.0102	0.0811 ± 0.0102
MCF _{low}	0.0812 ± 0.042	0.306 ± 0.024	0.1505 ± 0.005	0.1931 ± 0.016	0.1825 ± 0.0143	0.0734 ± 0.0093
CCGAIN	0.0673 ± 0.011	0.2330 ± 0.018	0.1402 ± 0.001	0.1091 ± 0.014	0.1024 ± 0.0175	0.0425 ± 0.0096

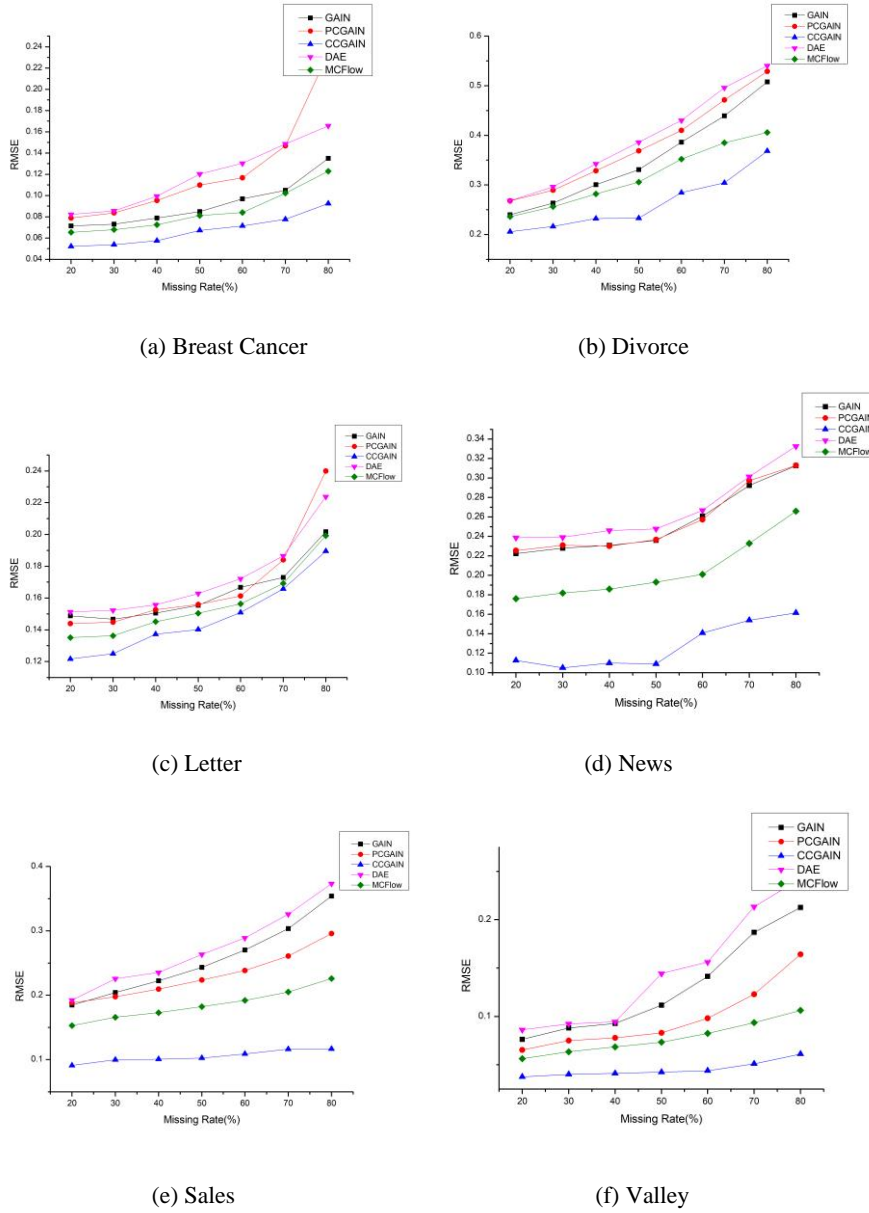


Fig. 5. RMSE of CCGAIN versus other methods with various missing rates.

4.2 Imputation Accuracy in UCI Datasets

(1) We compare the RMSE values of data imputation results between CCGAIN and the methods EM, KNN, MissForest, MICE, DAE, GAIN, PCGAIN and MCFIow. A smaller RMSE value indicates higher imputation accuracy. In the experiments, we set the

percentage of missing data to 50% of the total data, and each experiment was repeated 10 times. The average of these ten experiments was taken as the final result. The comparison results of the experiments are shown in Table 2.

From Table 2, it is evident that the method proposed in this study exhibits the lowest RMSE (Root Mean Square Error) values across all datasets. Upon examining the experimental data, CCGAIN demonstrates a significant reduction in errors on datasets such as Divorce, Sales, and News. Although the improvement is relatively modest on datasets like Breast Cancer, Valley, and Letter, it still surpasses other methods. This indicates that compared to other baseline methods, the method proposed in this study yields smaller errors, allowing for more accurate imputation of missing data.

(2) RMSE values of imputation results by CCGAIN, GAIN, PCGAIN, DAE, and MCFlow are compared with varying levels of data missing rates, ranging from 20% to 80% with intervals of 10%. As shown in Fig. 5, where the blue line represents CCGAIN, it can be observed that CCGAIN consistently outperforms other methods across different levels of data missing rates. Particularly, the superiority of CCGAIN becomes more pronounced as the missing rate increases. This suggests that the method proposed in this study achieves more accurate imputation results compared to DAE, MCFlow, GAIN, and PCGAIN, especially when dealing with higher missing rates.

4.3 Prediction Performance under Various Missing Rates

The post-imputation prediction accuracy of imputed data will be compared. The study conducted classification predictions using the XGBoost classifier on imputed data from the Divorce and Breast Cancer datasets at missing rates of 20%, 40%, 60%, and 80%. As the proposed CCGAIN is an improvement over GAIN, we will compare CCGAIN with GAIN in this context.

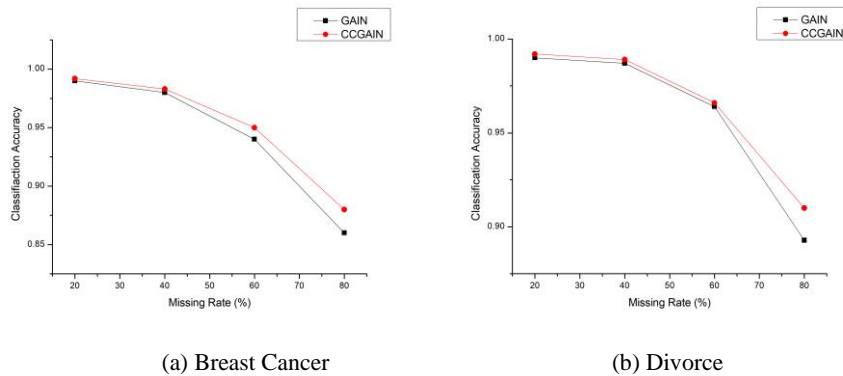


Fig. 6. Post-imputation prediction accuracy of CCGAIN versus GAIN with various missing rates.

The comparison results are depicted in Fig. 6, where the red line represents the classification prediction accuracy of data imputed by CCGAIN. It can be observed that the

classification accuracy of CCGAIN-imputed data is consistently equal to or greater than that of GAIN-imputed data across all missing rates. Moreover, the superiority of CCGAIN becomes more evident as the missing rate increases. This further validates that CCGAIN effectively preserves class information during imputation.

4.4 Ablation Experiment

We will conduct ablation experiments on CCGAIN. We will break down the operations targeting the two limitations of CCGAIN into two parts: the first is the clustering process, and the second is the construction of multi-scale data and the imputation process. Consequently, we will remove these two parts separately. One is to remove the clustering part from CCGAIN, directly construct multi-scale data and impute (CCGAIN w/o C). The other is to cluster the dataset but use GAIN for imputation within each cluster (CCGAIN w/o M). At a 50% data missing rate, CCGAIN will be compared with the aforementioned two models in terms of RMSE. The experimental results are presented in Table 3. As shown in the table, the RMSE of CCGAIN is the smallest.

Table 3. Ablation experiment

Algorithm	Breast Cancer	Divorce	Letter	News	Sales	Valley
GAIN	0.085 ±0.0045	0.3309 ±0.02	0.1554 ±0.0072	0.2361 ±0.0226	0.2406 ±0.0190	0.1022 ±0.0095
CCGAIN w/o C	0.0859 ±0.028	0.2407 ±0.0032	0.1459 ±0.0016	0.2394 ±0.0039	0.1839 ±0.0071	0.1859 ±0.0063
CCGAIN w/o M	0.1336 ±0.031	0.2942 ±0.0099	0.1378 ±0.0027	0.2362 ±0.0134	0.2238 ±0.0027	0.0865 ±0.0009
CCGAIN	0.0673 ±0.011	0.2330 ±0.018	0.1402 ±0.0016	0.1091 ±0.0149	0.1024 ±0.0175	0.0425 ±0.0096

5 Conclusion

The paper primarily investigates the issue of imputing missing data, proposing a new imputation algorithm called CCGAIN based on the work of Yoon et al.'s GAIN. Firstly, a clustering module is introduced to partition the dataset into multiple clusters, and then imputation is performed separately within these clusters. Since samples within each cluster exhibit higher correlation, CCGAIN's imputation process is more targeted. Subsequently, multiple scales are constructed for the data within each cluster, allowing the imputation results at local scales to supervise the imputation results at the global scale, thereby constructing the reconstruction loss of missing values. Based on the reconstruction loss of missing values, the reconstruction loss of non-missing values, and the adversarial loss, imputation is performed at the global level. Finally, the data from these

clusters are merged to form the final imputation result. Experimental results demonstrate the effectiveness of the proposed method.

Acknowledgments. The author would like to thank the editors and the anonymous reviewers for their valuable comments and suggestions to improve the paper. This work was supported by the National Natural Science Foundation of China (No.62076002, 61402005, 61972001), the Natural Science Foundation of Anhui Province, China (No.2008085MF194, 1308085QF114, 1908085MF188), the Higher Education Natural Science Foundation of Anhui Province, China (No. KJ2013A015).

References

1. Júnior, G.A.D.S. and Silva, A.M.D.: A Simple and Efficient Incremental Missing Data Imputation Method for Evolving Neo-fuzzy Network. *Evolving Systems* 12 (1), 1-20 (2021)
2. Ezzine, I. and Benhlima, L.: A study of handling missing data methods for big data. 2018 IEEE 5th International Congress on Information Science and Technology, 498-501. IEEE, Piscataway (2018)
3. Hackl, A., Zeindl, J. and Ehrlinger, L.: Four factors affecting missing data imputation. Proceedings of the 35th International Conference on Scientific and Statistical Database Management, 1-2. Association for Computing Machinery, New York (2023)
4. Muñoz, J., Efthimiou, O., Audigier, V., de Jong, V.M. and Debray, T.P.: Multiple imputation of incomplete multilevel data using Heckman selection models. *Statistics in medicine* 43(3), 514-53 (2024)
5. LeCun, Y., Bengio, Y. and Hinton, G.: Deep learning. *Nature* 521(7553): 436-444 (2015)
6. Yoon, J., Jordon, J. and Schaar, M.: GAIN: Missing Data Imputation using Generative Adversarial Nets. Proceedings of the 2018 International conference on machine learning, 5689-5698. Association for Computing Machinery, New York (2018)
7. Adhikari, D., Jiang, W., Zhan, J., He, Z., Rawat, D.B., Aickelin, U. and Khorshidi, H.A.: A comprehensive survey on imputation of missing data in internet of things. *ACM Computing Surveys* 55(7), 1-38 (2022)
8. Sivakani, R. and Ansari, G.A.: Imputation using machine learning techniques. 2020 4th International conference on computer, communication and signal processing (ICCCSP), 1-6. IEEE, Piscataway (2020)
9. Platias, C. and Petasis, G.: A Comparison of Machine Learning Methods for Data Imputation. In Proceedings of the 11th Hellenic Conference on Artificial Intelligence, 150–159. Association for Computing Machinery, New York (2020)
10. Berti-Équille, L., Harmouch, H., Naumann, F., Novelli, N. and Thirumuruganathan, S.: Discovery of Genuine Functional Dependencies from Relational Data with Missing Values. Proceedings of the VLDB Endowment 11 (8), 880–892 (2018)
11. Hudak, A.T., Crookston, N.L., Evans, J.S., Hall, D.E. and Falkowski, M.J.: Nearest neighbor imputation of species-level, plot-scale forest structure attributes from LiDAR data. *Remote Sensing of Environment* 112(5), 2232-2245 (2008)
12. Ali, M., Jung, L.T., Abdel-Aty, A.H., Abubakar, M.Y., Elhoseny, M. and Ali, I.: Semantic-k-NN algorithm: An enhanced version of traditional k-NN algorithm. *Expert Systems with Applications* 151, 113374 (2020)
13. White, I.R., Royston, P. and Wood, A.M.: Multiple imputation using chained equations: Issues and guidance for practice. *Statistics in Medicine* 30(4), 377-399 (2011)

14. Stekhoven, D.J. and Bühlmann, P.: MissForest—non-parametric missing value imputation for mixed-type data. *Bioinformatics* 28(1),112-118 (2012)
15. García-Laencina, P.J., Sancho-Gómez, J.L. and Figueiras-Vidal, A.R.: Pattern classification with missing data: A review. *Neural Computing and Applications* 19(2), 263-282 (2010)
16. Gondara, L. and Wang, K.: Mida: Multiple imputation using denoising autoencoders. *Advances in Knowledge Discovery and Data Mining: 22nd Pacific-Asia Conference*, 260-272. Springer-Verlag, Berlin (2018)
17. Nazabal, A., Olmos, P.M., Ghahramani, Z. and Valera, I.: Handling Incomplete Heterogeneous Data using VAEs . *Pattern Recognition* 107, 107501 (2020)
18. Spinelli, I., Scardapane, S. and Uncini, A.: Missing data imputation with adversarially-trained graph convolutional networks. *Neural Networks* 129, 249-260 (2020)
19. Lai, X., Wu, X., Zhang, L., Lu, W. and Zhong, C.: Imputations of missing values using a tracking-removed autoencoder trained with incomplete data. *Neurocomputing* 366(Nov.13), 54-65 (2019)
20. Nazabal, A., Olmos, P. M., Ghahramani, Z., & Valera, I.: Handling incomplete heterogeneous data using vaes. *Pattern Recognition* 107: 107501 (2020)
21. Richardson, T. W., Wu, W., Lin, L., Xu, B., & Bernal, E. A.: Mcflow: Monte carlo flow models for data imputation. *2020 IEEE/CVF conference on computer vision and pattern Recognition (CVPR)*, 14205-14214. IEEE, New York (2020)
22. Shahbazian, R. and Trubitsyna, I. DEGAIN: generative-adversarial-network-based missing data imputation. *Information* 13(12), 575 (2022)
23. Shahbazian, R. and Greco, S.: Generative Adversarial Networks Assist Missing Data Imputation: A Comprehensive Survey & Evaluation. *IEEE Access* (2023)
24. Awan, S. E., Bennamoun, M., Sohel, F., Sanfilippo, F., and Dwivedi, G.: Imputation of missing data with class imbalance using conditional generative adversarial networks. *Neurocomputing* 453: 164-171. (2021)
25. Yoon, S., Sull, S.: GAMIN: Generative adversarial multiple imputation network for highly missing data. *2020 IEEE/CVF conference on computer vision and pattern Recognition (CVPR)*, 8453–8461. IEEE, New York (2020)
26. Qiu, W., Huang, Y., & Li, Q.: IFGAN: missing value imputation using feature-specific generative adversarial networks. *2020 IEEE International Conference on Big Data (Big Data)*, 4715-4723. IEEE, New York (2020)
27. Li, S.C.X., Jiang, B. and Marlin, B.: Misgan: Learning from incomplete data with generative adversarial networks. *ArXiv: abs/1902.09599* (2019)
28. Zhang, C., Cui, Y., Han, Z., Zhou, J.T., Fu, H. and Hu, Q.: Deep partial multi-view learning. *IEEE transactions on pattern analysis and machine intelligence* 44(5), 2402-2415 (2020)
29. Wang, Y., Li, D., Li, X. and Yang, M.: PCGAIN: Pseudo-label conditional generative adversarial imputation networks for incomplete data. *Neural Networks* 141, 395-403 (2021)
30. Nagarajan, G. and Babu, L.D.: Missing data imputation on biomedical data using deeply learned clustering and L2 regularized regression based on symmetric uncertainty. *Artificial Intelligence in Medicine* 123, 102214 (2022)