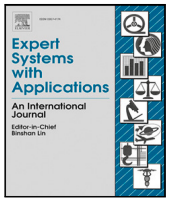
Expert Systems With Applications 227 (2023) 120201

Contents lists available at ScienceDirect

Expert Systems With Applications

journal homepage: www.elsevier.com/locate/eswa



Deep learning versus conventional methods for missing data imputation: A review and comparative study

Yige Sun [a], Jing Li [a], Yifan Xu [b], Tingting Zhang [c], Xiaofeng Wang [d,*]

[a] Department of Computer and Data Sciences, Case Western Reserve University, Cleveland, OH 44106, USA
[b] Meta Platforms, Inc., Menlo Park, CA 94025, USA
[c] Department of Statistics, University of Pittsburgh, Pittsburgh, PA 15260, USA
[d] Department of Quantitative Health Sciences, Cleveland Clinic, Cleveland, OH 44195, USA

ARTICLE INFO

Keywords:
Missing data imputation
Deep learning
Generative networks
MICE
MissForest

ABSTRACT

Deep learning models have been recently proposed in the applications of missing data imputation. In this paper, we review the popular statistical, machine learning, and deep learning approaches, and discuss the advantages and disadvantages of these methods. We conduct a comprehensive numerical study to compare the performance of several widely-used imputation methods for incomplete tabular (structured) data. Specifically, we compare the deep learning methods: generative adversarial imputation networks (GAIN) with onehot encoding, GAIN with embedding, variational auto-encoder (VAE) with onehot encoding, and VAE with embedding versus two conventional methods: multiple imputation by chained equations (MICE) and missForest. Seven real benchmark datasets and three simulated datasets are considered, including various scenarios with different feature types under different levels of sample sizes. The missing data are generated based on different missing ratios and three kinds of missing mechanisms: missing completely at random (MCAR), missing at random (MAR), and missing not at random (MNAR). Our experiments show that, for small or moderate sample sizes, the conventional methods establish better robustness and imputation performance than the deep learning methods. GAINs only perform well in the case of MCAR and often fail in the cases of MAR and MNAR. VAEs are easy to fall into mode collapse in all missing mechanisms. We conclude that the conventional methods, MICE and missForest, are preferable for practitioners to deal with missing data imputation for tabular data with a limited sample size (*i.e.*, $n < 30,000$) in real case analyses.

1. Introduction

Missing data commonly exist in a wide range of applications due to human errors, data processing issues, or cases that the relevant facts are not observed or not available. Missingness creates problems in data analyses and predictive modeling. Data imputation is an established practice to resolve the issue, i.e. estimating missing values from non-missing values in the dataset. Missing data imputation has been an active research area in both statistics and machine learning fields for a few decades (Rubin, 1976).

The validity and effectiveness of imputation strategies are affected by missing mechanism, formalized by Rubin and colleagues (Little & Rubin, 2019; Rubin, 1976), which describes the underlying mechanism that generates missing data that fall into three categories: missing completely at random (MCAR), missing at random (MAR), and missing not at random (MNAR). MCAR means that there is no relationship between missingness and either observed or unobserved variables. MCAR is a good practical start for imputation analysis because of its convenient hypothesis. However, it is not applicable to most real situations since actual missing generally involves complex relationships among observed variables and can be potentially affected by unobserved reasons. In contrast to MCAR, MAR occurs when the missingness is still random but depends on the observed variables. MAR is more general and more realistic than MCAR. Most statistical missing data imputation methods start from the MAR assumption. If neither MCAR nor MAR holds, we speak of MNAR. It means that the probability that an element is missing depends on the unobserved value of the missing elements. Specifically, MNAR happens if (1) the missing value influences the probability of missingness or (2) a certain unmeasured quantity predicts the value of the missing variable and/or the probability of missingness. For example, censored data in survival studies fall into this category. Rubin's

* Correspondence to: Xiaofeng Wang, Ph.D., Department of Quantitative Health Sciences, Lerner Research Institute, Cleveland Clinic, 9500 Euclid Ave/JJN3, Cleveland, OH 44195, USA.
E-mail addresses: yxs871@case.edu (Y. Sun), jingli@case.edu (J. Li), ethan.yifanxu@gmail.com (Y. Xu), tiz67@pitt.edu (T. Zhang), wangx6@ccf.org (X. Wang).

https://doi.org/10.1016/j.eswa.2023.120201
Received 6 August 2022; Received in revised form 15 March 2023; Accepted 15 April 2023
Available online 26 April 2023

distinction of these missing mechanisms is important for understanding why some methods will work, and others not.

Many methods and strategies have been developed to address the missing data problem. Generally, we can classify the available techniques into three categories: conventional statistical methods, machine learning methods, and (newly-developed) deep learning methods. Popular statistical methods include mean/median imputation, regression imputation, and expectation–maximization (EM). These methods are single imputation approaches, which are simple but do not account for the uncertainty of the prediction of missing values. Multiple Imputation was introduced by Rubin (1978) and refined in Little and Rubin (2019), Rubin (2004). It is a powerful imputation method that originates from a Bayesian analysis of a large-scale survey. Multiple imputation first creates several copies of the data set, each containing different imputed values. The imputation is then carried out on each data set using the same procedures. Finally, analyzing each data set separately yields multiple sets of parameter estimates and standard errors, and these multiple sets of results are combined into a single set of results. Multivariate imputation by chained equations (MICE) (Buuren & Groothuis-Oudshoorn, 2011), sometimes called "fully conditional specification", is one of the most popular and robust methods of multiple imputation. Machine learning-based imputation methods have been proposed and studied; the well-known machine learning methods include K-nearest neighbors (Batista & Monard, 2002), traditional feed-forward neural network (Gupta & Lam, 1996; Sharpe & Solly, 1995), and MissForest (Stekhoven & Bühlmann, 2012), and others. Many machine learning methods involve creating a predictive model to estimate missing values from the available information in the dataset. For example, missForest is an imputation method based on a random forest. By averaging over many unpruned classification or regression trees, missForest intrinsically constitutes a multiple imputation scheme.

In recent years, advances in deep learning models have motivated a suite of new imputation methods. Generative adversarial imputation nets (GAIN) (Yoon, Jordon, & Schaar, 2018) was proposed for missing data imputation using generative adversarial network (GAN). A method of multiple imputation using denoising auto-encoders (MIDA) was built from a de-noised auto-encoder (Gondara & Wang, 2018; Lu, Perrone, & Unpingco, 2020; Vincent, Larochelle, Bengio, & Manzagol, 2008). Several algorithms based on variational auto-encoders (VAE) were also suggested for imputation (Camino, Hammerschmidt, & State, 2019; McCoy, Kroon, & Auret, 2018; Qiu, Zheng, & Gevaert, 2020). Deep ladder imputation network (DLIN) is a novel deep learning imputation method, which incorporates the advantages of denoising auto-encoders and ladder architecture into an innovative formulation (Hallaji, Razavi-Far, & Saif, 2021). Although most of these deep learning methods demonstrated improved performances over traditional methods in certain simulation settings, there is no comprehensive comparison of these state-of-the-art methods versus conventional statistical and machine learning methods for data with different types of variables under different missing mechanisms.

This paper reviews the popular statistical, machine learning, and deep learning approaches, and discusses the advantages and disadvantages of these methods. A comparative numerical study is conducted to compare various methods under different scenarios. GAIN and VAE are chosen as the representatives for deep learning methods to compare with the widely-used statistical method, MICE, and the machine learning method, missForest. The two selected conventional methods have been shown that they outperform other common imputation techniques in terms of imputation error and maintenance of predictive ability (Shah, Bartlett, Carpenter, Nicholas, & Hemingway, 2014; Waljee et al., 2013). We design numerical experiments to compare the aforementioned methods in comprehensive settings by varying the types of data sets (continuous, categorical, and mixed-type), the missing mechanisms, the levels of correlation, and the missing ratios.

The rest of the paper is organized as follows. Section 2 reviews the popular statistical, machine learning, and deep learning based methods. Section 3 presents the setting of the numerical experiments to compare the performances of the recently developed deep learning methods, GAIN and VAE, versus the widely used and accepted methods, MICE and missForest. Section 4 summarizes the results of the numerical study. Section 5 provides a discussion of our findings. The advantages and disadvantages of the different missing data imputation approaches are addressed. Concluding remarks and recommendations are given in Section 6. All software code files and example datasets in this study are available at the GitHub through the link: https://github.com/EagerSun/A-comparison-of-Deep-Learning-and-conventional-statistical-methods-for-missing-data-imputation.

2. Review of missing data imputation methods

2.1. Problem definition

Assume that a d-dimensional random variable $\boldsymbol{X} = (X_1, \dots, X_d)$ (continuous or categorical) follows a probability distribution $p(\boldsymbol{X})$, and there is a d-dimensional variable $\boldsymbol{M} = (M_1, \dots, M_d)$ taking value in $\{0,1\}^d$. We call $\boldsymbol{M}$ the *mask* variable. We further define $\boldsymbol{X}^M = (X_1^M, \dots, X_d^M)$ such that

$$X_k^M = \begin{cases} X_k, & \text{if } M_k = 1 \\ \varnothing, & \text{if } M_k = 0 \end{cases}, \quad k = 1, \dots, d. \tag{1}$$

Here $\varnothing$ represents an unobserved value. The mask variable $\boldsymbol{M}$ is an indicator variable that indicates which elements of $\boldsymbol{X}$ are missing. In the missing data imputation problem, we observe a sample $\{\boldsymbol{x}_1^M, \dots, \boldsymbol{x}_n^M\}$ and $\{\boldsymbol{m}_1, \dots, \boldsymbol{m}_n\}$. The goal is to impute the unobserved values in $\{\boldsymbol{x}_1^M, \dots, \boldsymbol{x}_n^M\}$.

2.2. Conventional statistical methods

2.2.1. Mean/median/mode imputation

For a continuous variable, mean/median imputation is a simple method in which the mean or median of the observed values for each variable is computed and the missing values for that variable are imputed by this mean or median. For a categorical variable, the missing values for that variable are imputed by the mode of the observed values.

2.2.2. Regression imputation

Regression imputation assumes a linear relationship between variables. It assumes that the value of one variable changes in a linear way with the other variables. Mean imputation can be viewed as the simplest application of regression imputation. Regression imputation generally consists of two steps: a linear regression model is estimated on the basis of observed values in the target variable and some explanatory variables; the model is then used to predict values for the missing cases in the target variable. Missing values of the variable are replaced based on these predictions.

There are types of regression imputation: (1) deterministic regression imputation — it replaces missing values with the exact prediction of the regression model. Random variation around the regression slope is not considered, hence, imputed values are often too precise; and (2) stochastic regression imputation — it adds an additional random error term to the predicted value imputed by deterministic regression imputation, which solves the above issue.

2.2.3. Expectation–maximization

The expectation–maximization (EM) algorithm is a general method for obtaining maximum likelihood estimates when data are missing (Dempster, Laird, & Rubin, 1977). It is an iterative procedure

in which it uses other variables to impute a value (Expectation), then checks whether that is the value most likely (Maximization). Each iteration includes two steps: (1) The expectation step (E-step) uses the current estimate of the parameter to impute (expectation of) missing data, (2) The maximization step (M-step) uses the updated data from the E-step to find a maximum likelihood estimate of the parameter. The iterative process continues until there is convergence in the parameter estimates. EM imputation is generally better than mean/median imputation since it preserves the relationship with other variables.

2.2.4. MICE

Multiple imputation has several advantages over the above single imputation approaches. It involves filling in the missing values multiple times by creating multiple imputed datasets. The analyses of multiple imputed datasets take the uncertainty of imputation into account and yield the standard errors of estimates. MICE is one of the most popular multiple imputation techniques (Buuren & Groothuis-Oudshoorn, 2011). It operates under the assumption that the missing data are MAR.

We assume that the multivariate distribution $p(\boldsymbol{X}|\boldsymbol{\theta})$ of $\boldsymbol{X}$ is completely specified by $\boldsymbol{\theta}$, a vector of unknown parameters. The problem is how to estimate the distribution of $\boldsymbol{\theta}$. The MICE algorithm is a Gibbs sampler that estimates the posterior distribution of $\boldsymbol{\theta}$ by sampling iteratively from conditional distributions: $p(X|X_{-1},\boldsymbol{\theta}_1)$, …, $p(X|X_{-k},\boldsymbol{\theta}_k)$, …, $p(X|X_{-d},\boldsymbol{\theta}_d)$, where $X_{-k}=(X_1,\dots,X_{k-1},X_{k+1},\dots,X_d)$ denotes the collection of the $d-1$ variables in $\boldsymbol{X}$ except X_k, and the parameters $\boldsymbol{\theta}_1,\dots,\boldsymbol{\theta}_d$ are specific to the corresponding conditional densities.

In conventional applications of the Gibbs sampler, the full conditional distributions are derived from the joint probability distribution. However, in MICE, the joint distribution is only implicitly known and may not actually exist, so the conditional densities are not necessarily the product of a factorization of the joint distribution $p(\boldsymbol{X}|\boldsymbol{\theta})$. Starting from a simple draw from observed marginal distributions, the Gibbs sampler successively draws

$$\hat{\boldsymbol{\theta}}_1^{(t)} \sim p(\boldsymbol{\theta}_1|X_1^{obs},X_2^{(t-1)},\dots,X_d^{(t-1)})$$
$$\hat{X}_1^{(t)} \sim p(X_1|X_1^{obs},X_2^{(t-1)},\dots,X_d^{(t-1)},\hat{\boldsymbol{\theta}}_1^{(t)})$$
$$\dots$$
$$\hat{\boldsymbol{\theta}}_d^{(t)} \sim p(\boldsymbol{\theta}_d|X_d^{obs},X_1^{(t)},\dots,X_{d-1}^{(t)})$$
$$\hat{X}_d^{(t)} \sim p(X_d|X_d^{obs},X_1^{(t)},\dots,X_d^{(t)},\hat{\boldsymbol{\theta}}_d^{(t)})$$

where X_k^{obs} and $\hat{X}_k^{(t)}$ stand for the observed and imputed data for the kth variable at iteration t, and $X_k^{(t)}=(X_k^{obs},\hat{X}_k^{(t)})$. The convergence of this algorithm is typically fast.

Buuren and Groothuis-Oudshoorn (2011) presented an R package `mice`, which extends the standard algorithm of MICE in several ways. The new functionalities include imputing multilevel data, automatic predictor selection, post-processing imputed values, specialized pooling routines, model selection tools, and diagnostic graphs.

2.3. Machine learning methods

2.3.1. K-Nearest neighbors

A K-nearest neighbor model imputes missing values using only similar cases in a dataset. The method finds the K samples in the dataset "closest" to an incomplete data point and then averages the K data points to fill in the missing value. Typically, a K-nearest neighbor method uses a distance metric to compute the nearest neighbors. Configuration of K-nearest neighbor imputation often involves selecting the distance measure (e.g. Euclidean) and the number of contributing neighbors for each prediction, the hyperparameter K of the algorithm. The optimal value of K is usually chosen by cross-validation. The K-nearest neighbor method can impute both continuous variables (the mean or weighted mean among the K-nearest neighbors) and categorical variables (the mode among the K-nearest neighbors).

2.3.2. Feed-forward neural network

Traditional feed-forward neural network modeling can be used to estimate missing values by training a network to learn the incomplete variable (as an output) using the remaining complete variables as inputs. Sharpe and Solly (1995) proposed the following imputation scheme: (1) Given a dataset containing missing values, let us denote the subset that has missing values as $\boldsymbol{X}^{\varnothing}$ and the subset that does not contain any missing values as $\boldsymbol{X}^{c}$. For each possible combination of incomplete variables in $\boldsymbol{X}^{\varnothing}$, one constructs a feed-forward neural network using $\boldsymbol{X}^{c}$. Depending on the type of variable to be imputed (numerical or categorical), different type of error function is used during the training process. (2) After each neural network model is trained, unknown values are predicted using the corresponding model. The imputed values are obtained by averaging the missing data estimates by each model.

2.3.3. MissForest

Stekhoven and Bühlmann (2012) proposed an iterative imputation method called "missForest" that is based on random forests (RF). In each iteration, to impute missing values of a variable X_k^M, missForest first fits a RF with $X_k^M \sim \boldsymbol{X}_{-k}^M$ using rows that do not contain missing X_k^M values, then it uses the trained RF to predict missing values in X_k^M. This process is done for all $k \in \{1,\dots,d\}$, in the order of the number missing values in X_k^M from the fewest to the most. For the first iteration, missForest makes an initial guess for the missing values in $\boldsymbol{X}^M$ using mean imputation or another simple imputation method. The whole procedure is repeated until a stopping criterion is met.

By averaging over many unpruned classification or regression trees, RF intrinsically constitutes a multiple imputation scheme. Using the built-in out-of-bag error estimates of random forest, one can estimate the imputation error without the need of a test set. In Stekhoven and Buhlmann's study, missForest outperforms other imputation methods, including K-nearest neighbors. It performs especially well in data settings where complex interactions and non-linear relations are suspected.

2.4. Deep learning methods

2.4.1. GAIN

Based on the mechanism of GAN Goodfellow et al. (2020), Yoon et al. (2018) proposed a generative model framework for missing data imputation, named GAIN. We outline the procedure below. GAIN consists of two main components: (1) A generator G that imputes the missing data conditioned on the observed data and outputs a completed vector, and (2) A discriminator D that takes the output of G and attempts to identify which parts of the data are imputed. To ensure that G produces a unique distribution, the authors introduced a hint vector that is correlated to the missing pattern to D.

Fig. 1 shows the data flow of GAIN. The input of the generator, G, consists of three components $\boldsymbol{X}^M$, $\boldsymbol{M}$ and $\boldsymbol{R}$, where $\boldsymbol{X}^M$ and $\boldsymbol{M}$ are defined in 2.1, and $\boldsymbol{R}=\{R_1,\dots,R_d\}$ is d-dimensional noise variable, which is independent of $\boldsymbol{X}^M$ and $\boldsymbol{M}$. The generator, G, imputes missing values and produces an output variable $\hat{\boldsymbol{X}}$, where the missing values in $\boldsymbol{X}$ are replaced by the imputed values. Then we have

$$\hat{\boldsymbol{X}} = \boldsymbol{M}\odot\boldsymbol{X}^M + (1-\boldsymbol{M})\odot G(\boldsymbol{X}^M,\boldsymbol{M},(1-\boldsymbol{M})\odot\boldsymbol{R}) \tag{2}$$

where $\odot$ denotes the element-wise multiplication.

The inputs of the discriminator in GAIN, D, consist of two components, $\hat{\boldsymbol{X}}$ and $\boldsymbol{H}$, where $\boldsymbol{H}$ is a hint variable of the same shape as $\boldsymbol{M}$ and is obtained by randomly selecting a certain portion of values in $\boldsymbol{M}$ to reveal to D. The output of D is the predicted mask, $\hat{\boldsymbol{M}}$, which is calculated as follows:

$$\hat{\boldsymbol{M}} = D(\hat{\boldsymbol{X}},\boldsymbol{H}) \tag{3}$$

where element $\hat{m}_{ij} \in \hat{\boldsymbol{M}}$ indicates the likelihood that the ijth value in $\hat{\boldsymbol{X}}$ is an observed (non-missing) data. By modifying the degree of

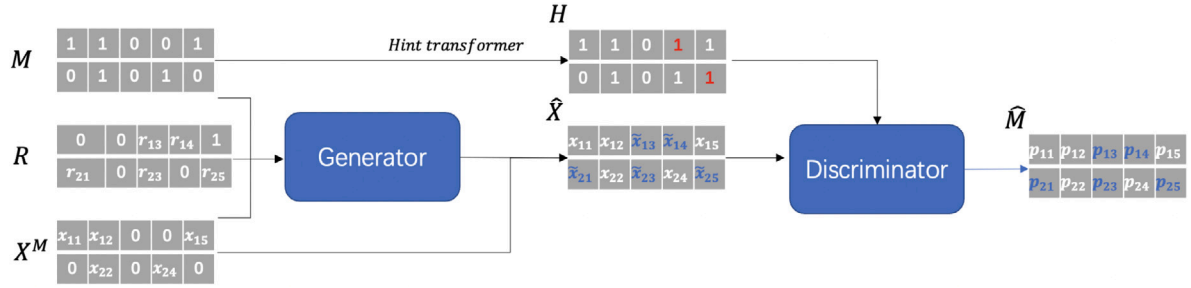


Fig. 1. The data flow of GAIN.

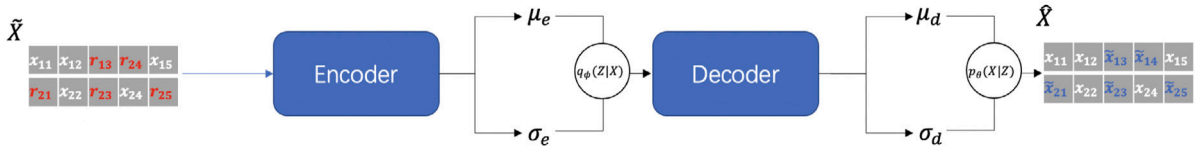


Fig. 2. The dataflow of VAE for imputation.

correlation between $\boldsymbol{H}$ and $\boldsymbol{M}$, we control the amount of "hint" to pass to D.

Similar to GAN, The loss function of GAIN is defined as:

$$Q(G, D) = \mathbb{E}_{\hat{\boldsymbol{X}},\boldsymbol{M},\boldsymbol{H}}\left[\boldsymbol{M}^T \log(\hat{\boldsymbol{M}}) + (1-\boldsymbol{M})^T \log(1-\hat{\boldsymbol{M}})\right]$$

where log is the element-wise logarithm function, and $\hat{\boldsymbol{X}}$ and $\hat{\boldsymbol{M}}$ are defined in (2) and (3). Thus, one is to solve the minimax problem,

$$\min_G \max_D Q(G, D).$$

2.4.2. VAE

VAE (Kingma & Welling, 2014) is another deep learning-based generative framework, which can be utilized for missing data imputation (Camino et al., 2019; McCoy et al., 2018). In the VAE framework, one assumes that the real data $\boldsymbol{X}$ is generated by a latent variable $\boldsymbol{Z}$ that follows a multivariate Gaussian distribution. An encoder generates the posterior distribution $q_\phi(\boldsymbol{Z}|\boldsymbol{X}) \sim \mathcal{N}(\boldsymbol{\mu}_e, \boldsymbol{\sigma}_e)$ of $\boldsymbol{Z}$ given $\boldsymbol{X}$, while a decoder generates the distribution $p_\theta(\boldsymbol{X}|\boldsymbol{Z}) \sim \mathcal{N}(\boldsymbol{\mu}_d, \boldsymbol{\sigma}_d)$ of $\boldsymbol{X}$ given $\boldsymbol{Z}$. Both the encoder and the decoder are neural networks, parameterized by ϕ and θ, respectively. Note that the input to the decoder is a random sample drawn from $q_\phi(\boldsymbol{Z}|\boldsymbol{X})$. The reconstructed $\boldsymbol{X}$ is a random sample drawn from $p_\theta(\boldsymbol{X}|\boldsymbol{Z})$. The parameters ϕ and θ are learned by maximizing the evidence lower bound (ELBO),

$$\text{ELBO} = E_{q_\phi(\boldsymbol{Z}|\boldsymbol{X})}\left[log\left(p_\theta(\boldsymbol{X}|\boldsymbol{Z})\right)\right] - KL\left[q_\phi(\boldsymbol{Z}|\boldsymbol{X})\|p(\boldsymbol{Z})\right] \quad (4)$$

where $p(\boldsymbol{Z})$ is the prior distribution of $\boldsymbol{Z}$, assumed to follow the standard Gaussian distribution. Maximizing the first term in (4) is equivalent to minimizing the reconstruction loss. The second term can be regarded as a regularization term. The readers are referred to Kingma and Welling (2014) for the full derivation of the equations.

Fig. 2 shows the dataflow for imputing missing values of $\boldsymbol{X}^M$ using VAE. First, we train a VAE where missing values in $\boldsymbol{X}^M$ are filled with random values, which we denote it as $\tilde{\boldsymbol{X}}$. The reconstruction loss in (4) is calculated over only non-missing values during training. Then we iteratively pass $\tilde{\boldsymbol{X}}$ through the trained VAE, replacing missing values with the reconstructed values from the previous iteration, until the reconstructive loss over non-missing values falls below a predetermined threshold. The reconstructed missing values in the last iteration are the imputed values. More specifically,

$$\begin{cases} \boldsymbol{X}^{imp} \sim \boldsymbol{p}_\theta(\boldsymbol{X}|\boldsymbol{Z}) = \mathcal{N}(\boldsymbol{\mu}_d, \boldsymbol{\sigma}_d) \\ \hat{\boldsymbol{X}} = \boldsymbol{M} \odot \boldsymbol{X}^M + (1-\boldsymbol{M}) \odot \boldsymbol{X}^{imp} \end{cases}$$

where $\boldsymbol{X}^{imp}$ denotes the imputed data vector sampled from complete distribution $p_\theta(\boldsymbol{X}|\boldsymbol{Z})$ and $\hat{\boldsymbol{X}}$ represents the resulting complete data vector.

2.5. DLIN

DLIN is a novel deep learning method, whose architecture is different than GAIN and VAE (Hallaji et al., 2021). It incorporates the denoising auto-encoders and ladder network into a unified framework. The deep ladder network is a relatively new approach to semi-supervised learning that turned out to be very successful (Rasmus, Berglund, Honkala, Valpola, & Raiko, 2015). In a standard autoencoder network, all information has to go through the highest layer. It needs to represent all the details of the input x. The intermediate hidden layer output $h^{(l)}$ in the network cannot independently represent information because they only receive information from the highest layer. In contrast, lateral connections at each layer in a deep ladder network give a chance for each $h^{(l)}$ to represent information from the higher layers.

Most deep learning-based imputation methods cannot utilize unobserved data to train an imputation model. However, ladder networks built in a DLIN can be used to remove this limitation. In a DLIN model, one first gives initial imputed data, $\boldsymbol{X}_0$, and then generates a stochastically corrupted version, $\check{\boldsymbol{X}}$. Consider that a DLIN has L hidden layers. The autoencoder is formulated by

$$\mathcal{E}(\check{x}_i) = (\check{x}_i, z_1, z_2, \dots, z_L, \hat{x}_i),$$

where z_l is the latent variable (hidden representation) at the lth hidden layer, and $\hat{x}_i$ is an estimate of $\check{x}_i$, and the autoencoder decoder network is modeled as:

$$\mathcal{E}_c(\check{x}_i) = (\check{x}_i, z_1^c, z_2^c, \dots, z_L^c, \hat{x}_i^c)|\boldsymbol{m}_i = \boldsymbol{1},$$

$$\mathcal{D}(\check{z}_i) = (z_L^c, z_1^d, z_2^d, \dots, z_L^d)|z_l^c \in \mathcal{E}_c(\check{x}_i),$$

where $\boldsymbol{m}_i$ is the ith mask variable defined in Section 2.1, $\boldsymbol{1}$ is a $1 \times n$ vector of ones. z_l^c and z_l^d are the hidden representations of $\mathcal{E}_c$ and $\mathcal{D}$ at the lth hidden layer, respectively. $\hat{x}_i$ is an estimate of $\check{x}_i$. In a DLIN, a fusion function is used to fuse the noisy representations in $\mathcal{E}_c$ and the denoised representations through the lateral connections. Thus, the model facilitates the generation of more abstract features at each layer of the ladder network (Hallaji et al., 2021). It has been shown that DLIN performs well for data imputation with high missing ratios. It also can handle cases where temporal and/or spatial missing values exist among the data.

3. Numerical experiments

The efficacy of the missing data imputation methods depends heavily on the problem domain, for example, sample size, types of variables,

Table 1
Summary of the ten datasets in the numerical study.

Dataset	Summary	Characteristics
Continuous data		
PimaIndiansDiabetes	$n = 768$, $d = 8$. It was from a study of Pima Indians' Diabetes. It consists of several medical variables including the number of pregnancies the patient has had, their BMI, insulin level, age, etc.	A typical moderate-size clinical data
Spam	$n = 4601$, $d = 57$. It was a dataset that classifies e-mails as spam or non-spam. The first 48 variables contain the frequency of some keyword in the e-mail. The variables 49–54 indicate the frequency of certain characters. The variables 55–57 contain the average, longest and total run-length of capital letters.	Skewed data, proportional and rate variables
DTI	$n = 382$, $d = 93$. It contains fractional anisotropy tract profiles in diffusion tensor images for patients with multiple sclerosis.	Functional (time series) data
Categorical data		
Carcinoma	$n = 118$, $d = 7$. A dataset for diagnoses of carcinoma cancer on seven categorical features representing pathologist ratings.	A small categorical dataset
DNA	$n = 3,186$, $d = 180$. It contains 180 indicator binary variables for primate splice-junction gene sequences.	High-dimensional categorical variables
Mixed data		
NCbirths	$n = 1450$, $d = 13$. It contains data on a sample of birth records from the North Carolina State Center for Health and Environmental Statistics. It includes 5 continuous and 8 categorical variables.	A typical moderate-size data with mixed variables
VietNamI	$n = 27,765$, $d = 12$. It was a dataset of medical expenses in Vietnam from 1997. It includes 7 continuous and 5 categorical variables.	A big dataset with imbalanced and skewed variables
Simulated1	$n = 100$, $d = 15$. A simulated data with 10 continuous and 5 categorical variables.	A small set with known covariance structure
Simulated2	$n = 500$, $d = 15$. A simulated data with 10 continuous and 5 categorical variables.	A moderate set with known covariance structure
Simulated3	$n = 1000$, $d = 15$. A simulated data with 10 continuous and 5 categorical variables.	A large set with known covariance structure

missingness patterns. In this study, we compare the performances of popular deep learning methods, GAIN and VAE, versus the widely used and accepted methods, MICE and missForest. We consider seven real datasets and three synthetic datasets that represent typical cases in practice. These datasets include cases with small, moderate, and large sample sizes and cases with continuous variables, categorical variables, and mixed-typed (both continuous and categorical) variables. We generate missing data in the datasets by three types of missing mechanisms: MCAR, MAR, and MNAR and three levels of missing ratios: 10%, 30% and 50%. Table 1 provides the basic summary information of each dataset.

Model performances are evaluated based on (averaged) root mean squared error (RMSE) for continuous variables and Accuracy for categorical variables, respectively. Assume that we have p continuous variables and q categorical variables among all d-dimensional variables. The average RMSE is given as

$$RMSE = \frac{1}{p}\sum_{k=1}^{p}\sqrt{\frac{1}{n_k}\sum_{i=1}^{n_k}\left(\hat{x}_{ki} - x_{ki}\right)^2},$$

where $\hat{x}_{ki}$ is the ith imputed value, x_{ki} is the ith true observed value, and n_k is the total number of the missed observations for the kth continuous variables ($k = 1, \ldots, p$). The averaged accuracy is given as

$$accuracy = \frac{1}{q}\sum_{j=1}^{q}\frac{\#\ correct\ label\ for\ the\ jth\ variable}{n_j},$$

where n_j is the total number of missed observations for the jth categorical variables ($k = 1, \ldots, q$).

RMSE is calculated on scaled datasets to avoid disproportionate variable contributions. RMSE and Accuracy provide us with measures of the relative distance between the imputed dataset and the original complete dataset. For multiple imputation scenarios with K imputations, we have K values for RMSE and/or Accuracy per dataset and we use the average to evaluate the model performance.

We conducted each experiment 100 times. We report mean RMSE and/or Accuracy along with their standard deviations (SD) as the performance metrics.

3.1. Real datasets

We include 7 real datasets in this study: *PimaIndiansDiabetes* (Ripley, 2007; Wahba, Gu, Wang, & Chappell, 2018), *spam* (Hastie, Tibshirani, Friedman, & Friedman, 2009), and *DTI* (Goldsmith, Crainiceanu, Caffo, & Reich, 2012) only contain continuous features, *Carcinoma* (Agresti, 2003) and *DNA* (Noordewier, Towell, & Shavlik, 1991) only contain categorical features, and *NCbirths* (Cannon et al., 2018) and *VietNamI* (Cameron & Trivedi, 2005) contain both continuous and categorical features. Table 1 provides a summary of these datasets and the special characteristics of each set. The supplementary document provides further details of them.

3.2. Synthetic data construction

In addition to the seven real benchmark datasets, we generate three mixed-type synthetic datasets with three levels of sample size (see Table 1). The reason we consider the simulated datasets here is that we know explicitly the data distribution and the covariance structure of the data. All three simulated datasets have 15 features that are sampled from multivariate Gaussian distributions with the following means and covariance matrices:

$$\boldsymbol{\mu}_s = [0, 0, 0, 1, 1, 1, 2, 2, 3, 3, 0, 0, 0, 0, 0] \tag{5}$$

$$\boldsymbol{\sigma}_s = 0.5 \cdot \boldsymbol{I} + 0.5 \cdot \mathbf{1} \tag{6}$$

where $\boldsymbol{I}$ is an identity matrix and $\mathbf{1}$ is a matrix of 1's. We then transfer the last five features of each synthetic data to categorical variables by dichotomization. Specifically, categorical levels are assigned as follows

$$c_i = \begin{cases} c_0 & \textbf{if } v \in (-\infty, a_1] \\ c_1 & \textbf{if } v \in (a_1, a_2] \\ \ldots & \\ c_n & \textbf{if } v \in (a_n, +\infty) \end{cases} \tag{7}$$

where v is the continuous value and $t_i = [a_1, a_2, \ldots, a_n]$ are cutoff thresholds for feature c_i. We choose the cutoff thresholds for the last

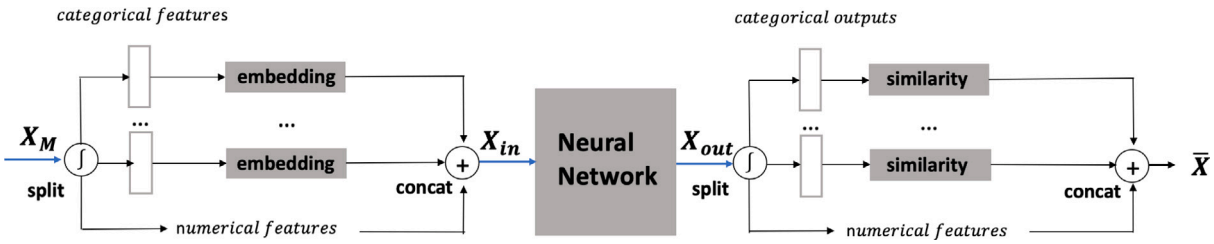


Fig. 3. The embedding structure of missing data $\boldsymbol{X}^M$.

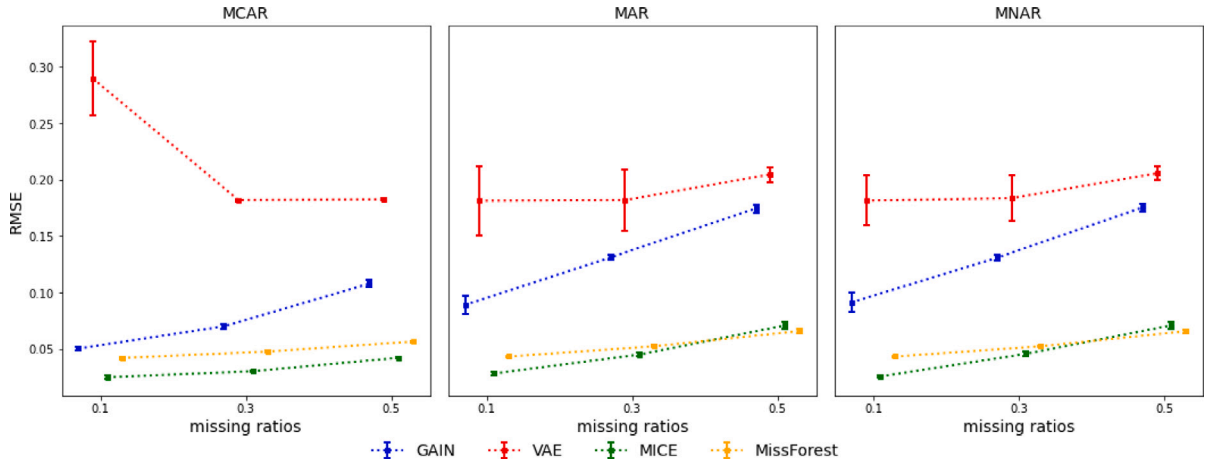


Fig. 4. The mean RMSEs with error bars of different methods under different missing ratios and different missing mechanisms for the DTI data.

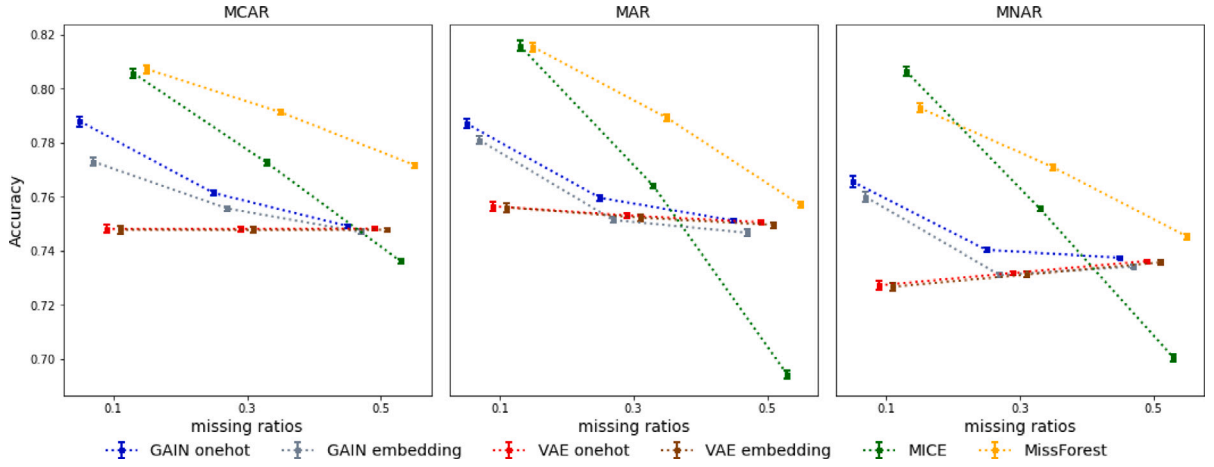


Fig. 5. The mean accuracies with error bars of different methods under different missing ratios and different missing mechanisms for the DNA data.

five features to be $t_1 = [-1, 0, 1]$, $t_2 = [-1, 0]$, $t_3 = [-0.5]$, $t_4 = [0]$, $t_5 = [0.5]$, respectively.

3.3. Generation of missing values

Let $\boldsymbol{M}$ with probability distribution p_M, parameterized by ϕ, denotes the missingness of data $\boldsymbol{X}$. Let $\boldsymbol{X}^{obs}, \boldsymbol{X}^{mis}$ denote the observed and missing values in $\boldsymbol{X}$, respectively. The three missing data generation procedures are described as follows.

3.3.1. MCAR

The probability that a value is missing does not depend on any other features, defined as the following:

$$p_M(\boldsymbol{M}|\boldsymbol{X}^{obs}, \boldsymbol{X}^{mis}) = p_M(\boldsymbol{M}). \tag{8}$$

3.3.2. MAR

The probability that a value of is missing only depends on observed values $\boldsymbol{X}^{obs}$. Defined as the following:

$$p_M(\boldsymbol{M}|\boldsymbol{X}^{obs}, \boldsymbol{X}^{mis}) = p_M(\boldsymbol{M}|\boldsymbol{X}^{obs}), \quad \forall \boldsymbol{X}^{obs}. \tag{9}$$

3.3.3. MNAR

Missing mechanisms that are neither MCAR nor MAR are considered MNAR. We create MNAR missing data by correlating ϕ with both $\boldsymbol{X}^{obs}$ and $\boldsymbol{X}^{mis}$.

We also consider another special scenario of MNAR, the bursty missing data. The "bursty" missing refers to the situation that the instances of missing data appear to be clustered in some groups/orders or at certain moments of time. Erhan et al. (2021) provided a great example to compare non-bursty missing and bursty missing data.

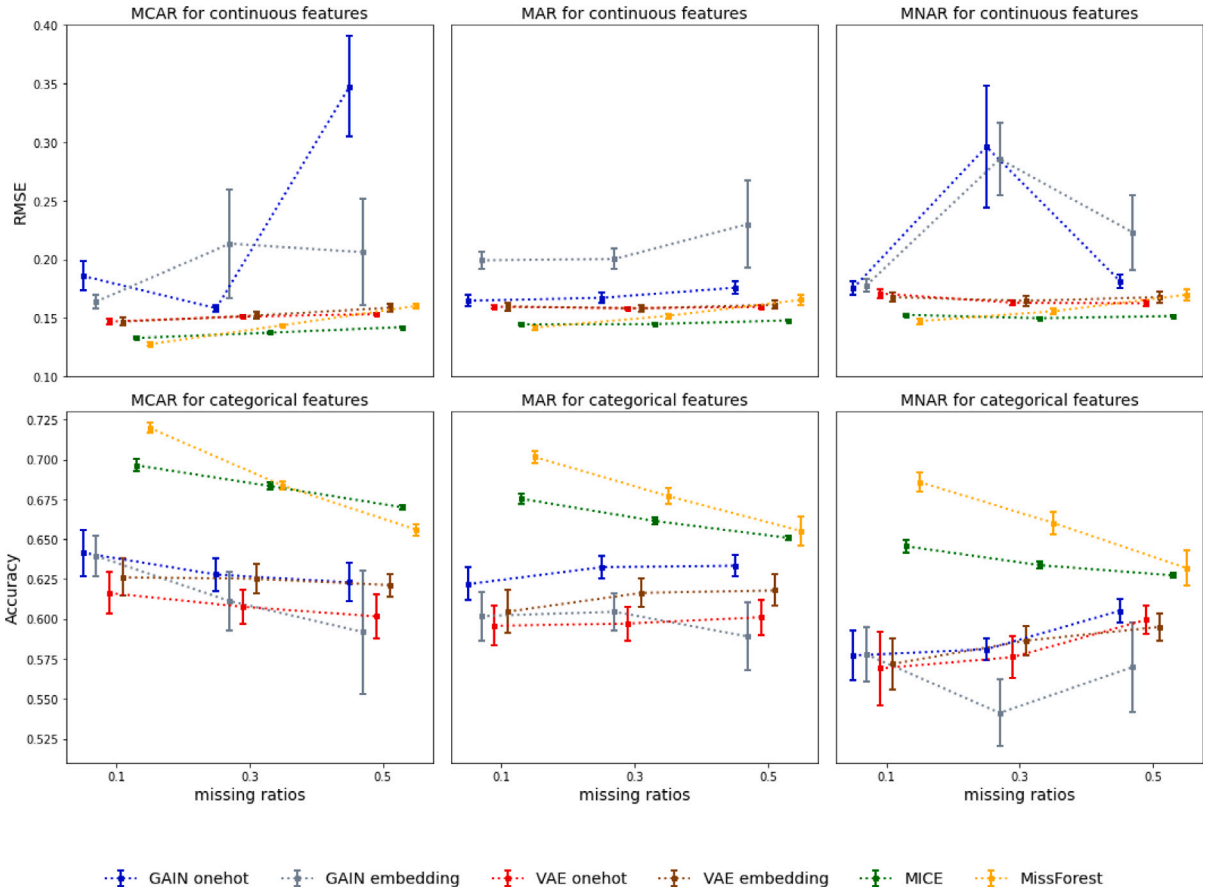


Fig. 6. The mean RMSEs with error bars (for continuous variables) and the mean accuracies with error bars (for categorical variables) of different methods under different missing ratios and different missing mechanisms for the VietNamI data.

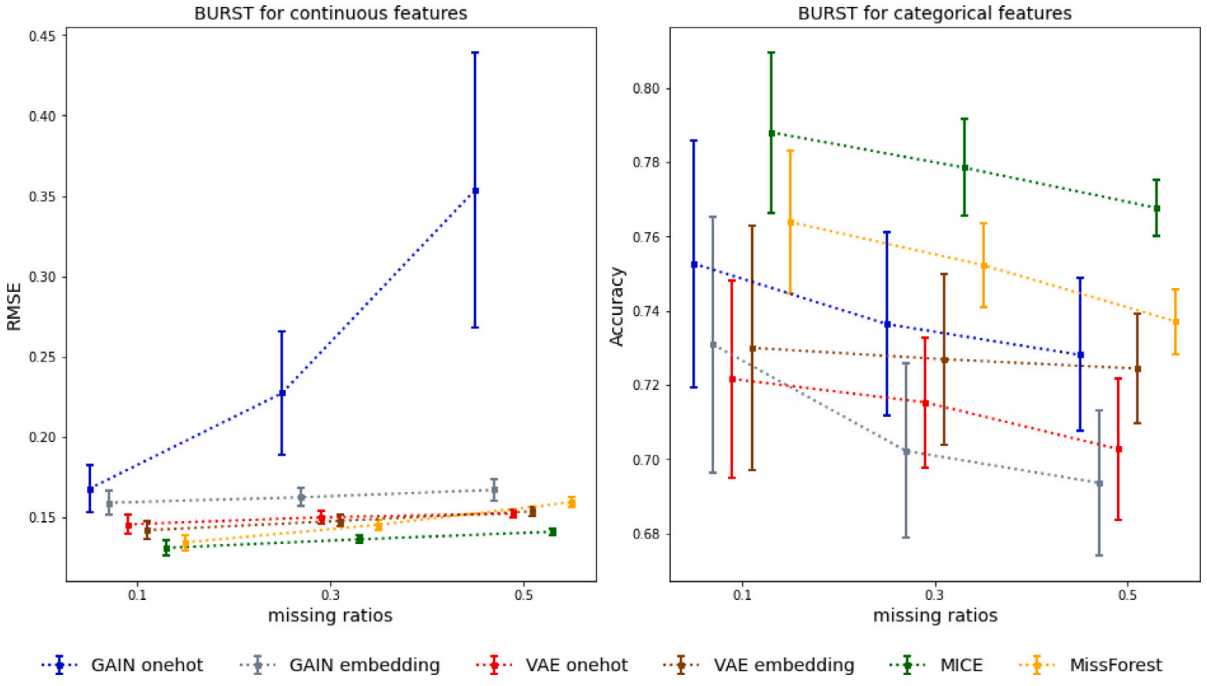


Fig. 7. The mean RMSEs with error bars (for continuous variables) and the mean accuracies with error bars (for categorical variables) of different methods under different missing ratios under burst-missing for the NCbirths data.

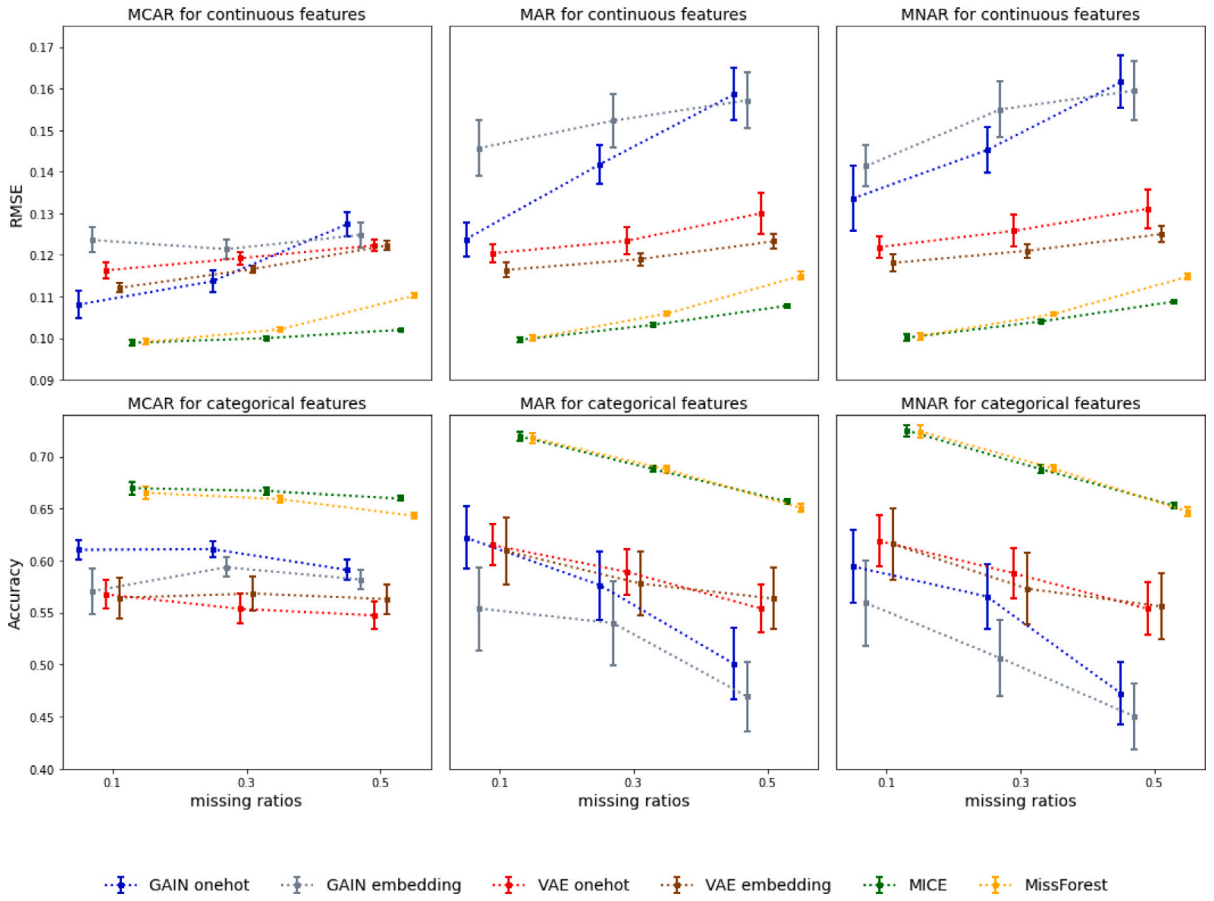


Fig. 8. The mean RMSEs with error bars (for continuous variables) and the mean accuracies with error bars (for categorical variables) of different methods under different missing ratios and different missing mechanisms for the simulated3 data.

3.4. Data preprocessing for deep generative models

Deep learning models cannot directly process categorical values. We apply onehot encoding and dense embedding to transform categorical values to numerical vectors.

- Onehot encoding: A categorical value with m possible unique categorical values $\boldsymbol{v} = \{v_1, \ldots, v_m\}$, is embedded as a vector $\boldsymbol{e} \in \{0,1\}^m$, where the entry corresponding to the index of the value in $\boldsymbol{v}$ is set to 1, and other to 0.
- Dense embedding : Each unique value is assigned to a numerical vector $\boldsymbol{e} \in \boldsymbol{R}^n$ that is updated during the training process. The aim of applying dense embeddings in deep generative models is to improve the imputation performance by learning the potential relations between features. The embedding vector of missing categorical values is filled with N/A's.

All categorical features are transformed into numerical vectors as illustrated in Fig. 3 for deep neural net-based methods. All observed continuous features are standardized into the range $[0,1]$ or $[-1,1]$ on the specific embedding method applied. Processed categorical and continuous features are concatenated to form the input feature vector for deep neural net-based models.

The imputed data from deep generative models are decoded back to the original space by data splitting: Values of continuous features are transferred to their original ranges. Embeddings of categorical features are mapped to the original value whose embedding is the closest according to cosine similarities.

3.5. Model training configurations

Conventional methods, e.g. MICE and missForest, are selected to compare with deep generative methods. Both methods are implemented in R using the packages `MICE` and `missForest`. For MICE, we apply predictive mean matching for continuous features, and logistic regression/multinomial regression for binary/multi-level categorical features, respectively. We set the number of multiple imputations to 50 for each incomplete dataset. Missing continuous values and categorical values are filled with the corresponding means and modes from those 50 imputation results. For missForest, we set the number of random trees for imputation to 200.

Deep generative methods, GAIN and VAE, are implemented in Python with Tensorflow/Keras (Abadi et al., 2015; Chollet et al., 2015). After the proper epochs and batch sizes of models under each miss-sampling setting are determined, we train GAIN and VAE with Adam/RMSprop optimization mechanism in the imputation process, respectively. Finally, we compute the result performance metrics from the trained models.

4. Results

4.1. Comparisons using real data

4.1.1. Pure continuous data

Fig. 4 shows the mean and standard deviations of RMSEs by different models under three missing mechanisms for the DTI data. Three levels of missing rates were considered: 10%, 30% and 50%. We observe that the conventional methods, MICE and missForest, outperform

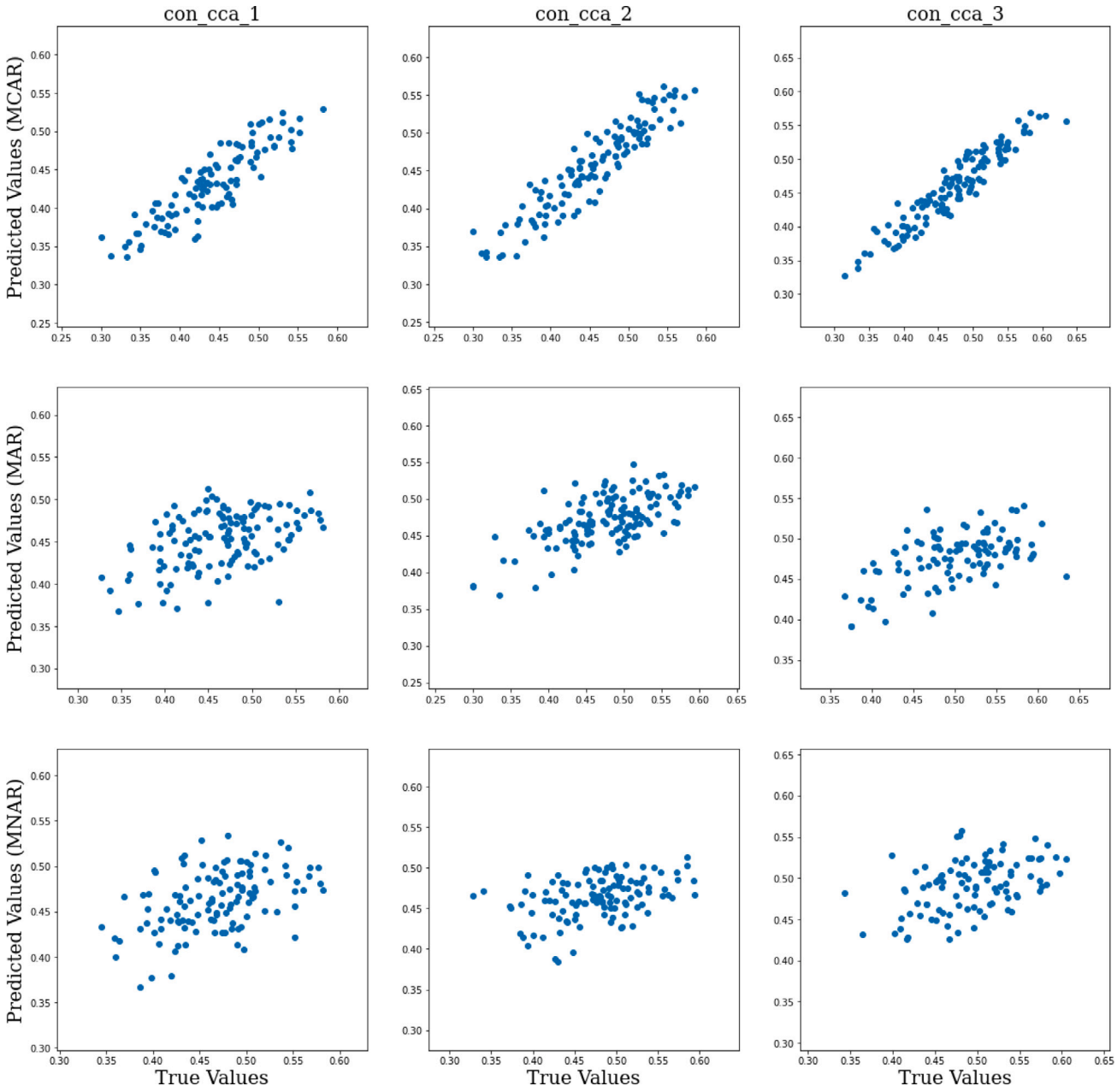


Fig. 9. The scatter plots of true v.s. predicted values of features in the study of DTI data (the case of 30% missing ratio) for GAIN: the first, second, and third row of each sub-figure correspond to the case of MCAR, MAR and MNAR, respectively. Three continuous features are shown (con_cat_1, con_cat_2 and con_cat_3).

the deep learning-based methods, GAIN and VAE. The VAE is the worst method in all scenarios. Unsurprisingly, the data with MCAR are easier to be imputed comparing the cases with MAR and MNAR. The RMSEs increase as the level of the missing rate increases. When the missing rate is low (e.g. 10% or 30%), MICE appears the winning among all methods. Note that the DTI data are functional (time-series) data, where the 93 variables are dependent (Ramsay & Silverman, 2005). Both MICE and missForest showed great performance for the functional dependent data. Table S-I in the supplement document presents all summarized results of RMSEs for two other real continuous datasets, PimaIndiansDiabetes and spam. PimaIndiansDiabetes is a typical clinical dataset, while spam contains skewed proportional/rate variables. The results for these cases also show similar findings as in Fig. 4, where both MICE and missForest are winners compared with deep learning methods.

4.1.2. Pure categorical data

Fig. 5 shows the mean and standard deviations of accuracies by different models under different missing rates and missing mechanisms for the DNA data. The DNA dataset has a large sample size ($N = 3186$) and high-dimensional variables ($d = 180$). We observe that MissForest is the winner among all methods. The performance of MICE is similar to that of MissForest as the missing ratio is small (ratio = 0.1). However, it is of interest to observe that MICE is the worst method as the missing ratio is large (ratio = 0.5). GAIN is better than VAE, but both the deep learning methods are worse than MissForest. We considered two types of encoding methods for deep learning imputation: onehot encoding and dense embedding. It appears that there is no obvious difference between the two encoding methods.

Table S-II in the supplement document shows the results of the other categorical dataset, Carcinoma. The sample size of Carcinoma is much smaller (n = 118). In this case, MICE is the winner among

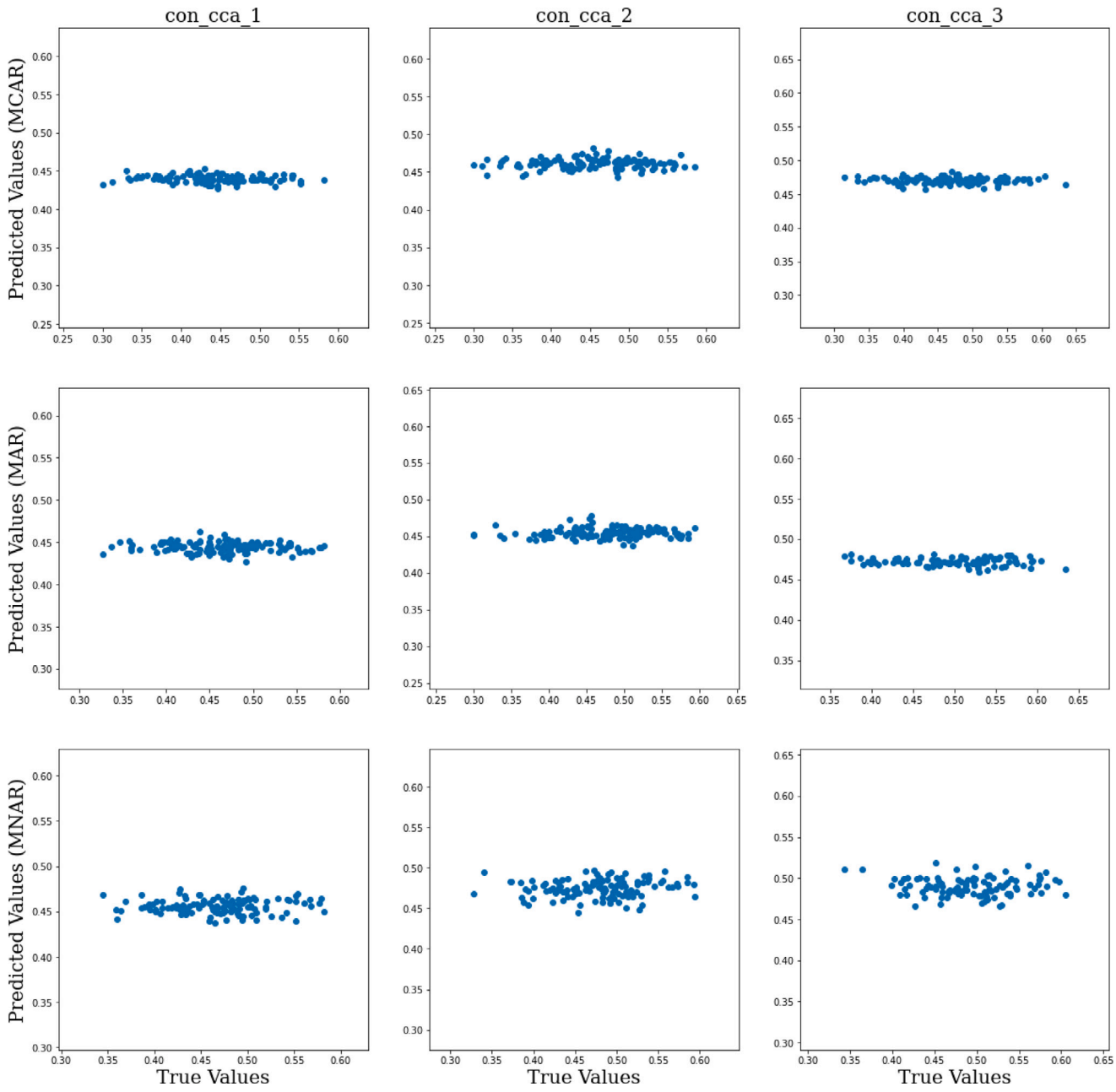


Fig. 10. The scatter plots of true v.s. predicted values of features in the study of DTI data (the case of 30% missing ratio) for VAE.

all methods. The performance of MissForest is close to that of MICE in all scenarios. They generally outperform deep generative models in imputing categorical variables.

4.1.3. Mixed data

Fig. 6 presents the results of the study of the ViteNamI data. This example dataset is a relatively big dataset with mixed variables. It contains 27,765 observations with 12 variables. We note that GAIN performs poorly in imputing missing values for the continuous variables. It shows unstable for three different missing mechanisms and different missing rates. For both continuous and categorical variables, the conventional methods are uniformly better than the deep generative models. Table S-III in the supplement document presents the summarized results for the other mixed data. The results show similar findings as in Fig. 6.

We also conducted a case study of the bursty missing mechanism, a special case of MNAR using the NCbirths data. Following the design in Erhan et al. (2021), we randomly select a corresponding number of bursts of a given block size (number of data points, here we set it as 50) to be invalidated from the dataset, in order to reach the desired dataset impairment level (10%, 30% and 50%). The results of the different imputation methods are shown in Fig. 7. Both MICE and missForest are better than the deep learning methods for the case of the bursty missing. GAIN with onehot seems not a good choice for bursty missing data with a high level of missing ratios (for example, 50%).

4.2. Comparisons using simulated data

As described in Section 3, the simulated data allow us to specify the known correlation structure among the variables. We considered here a compound-symmetry correlation. As summarized in Table S-IV, MICE and missForest outperform neural net-based methods on these simulated data as well. For the deep generative methods, GAIN is better than VAE in the case of MCAR, while VAE is better than GAIN in the

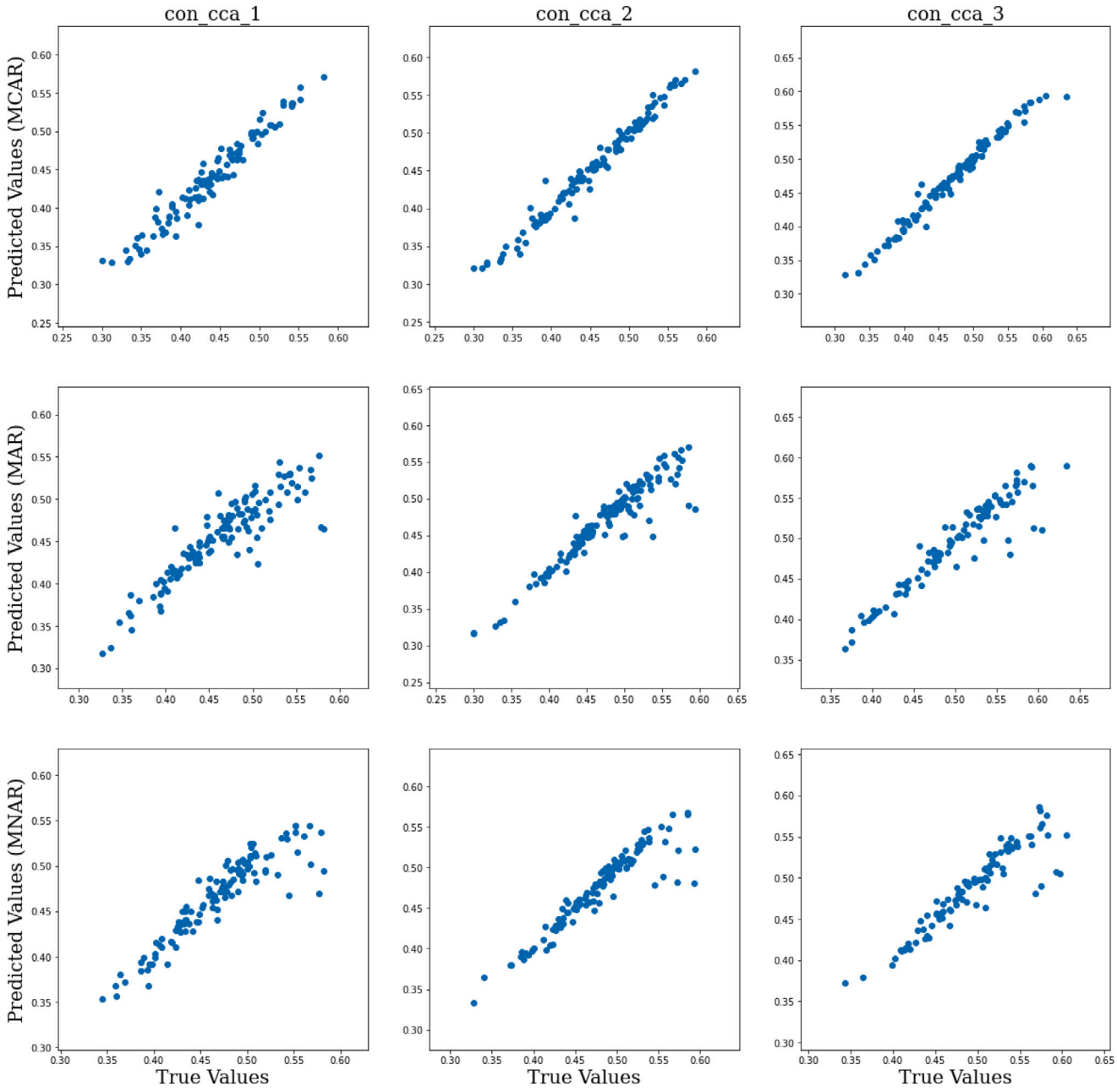


Fig. 11. The scatter plots of true v.s. predicted values of features in the study of DTI data (the case of 30% missing ratio) for MICE.

cases of MAR and MNAR. Comparing onehot encoding and embedding models, it is interesting that GAIN with embedding works worse than GAIN with onehot encoding. The VAE with embedding and the VAE with onehot encoding have similar performances in imputing both continuous and categorical variables. Fig. 8 displays the results for Simulated3, which shows that MICE and missForest are better than GAIN and VAE.

5. Discussion

We have performed a thorough investigation of practical imputation methods in the context of tabular data (i.e. structured data), investigating the performance of multiple missing data strategies using both real-world and simulated datasets. The scenarios were varied by using different sample sizes, missing data mechanisms, and ratios of missing data. In summary, we found that the conventional methods, MICE and missForest, outperform the recently proposed deep generative methods, GAIN and VAE, in these experiments. For the deep learning models, there is no obvious improvement using embedding v.s. onehot encoding for categorical features.

In our study, we also closely looked at model fitting through a graphical evaluation of the predicted values of missing data. Here we demonstrate a few examples to show that the deep learning models, including both GAIN and VAE, lead to a model collapse in many situations. Figs. 9–12 show the scatter plots of true versus predicted values of continuous features in the study of DTI data for the case of 30% missing ratio. We present three continuous features (con cat 1, con cat 2 and con cat 3). The first, second, and third rows correspond to the case of MCAR, MAR, and MNAR, respectively. Additional graphical examples can be found in the supplement document. Figure S-I presents the confusion matrices and the scatter plots of true versus predicted values for categorical and continuous features in the Simulated3 example for the case of 30% missing ratio.

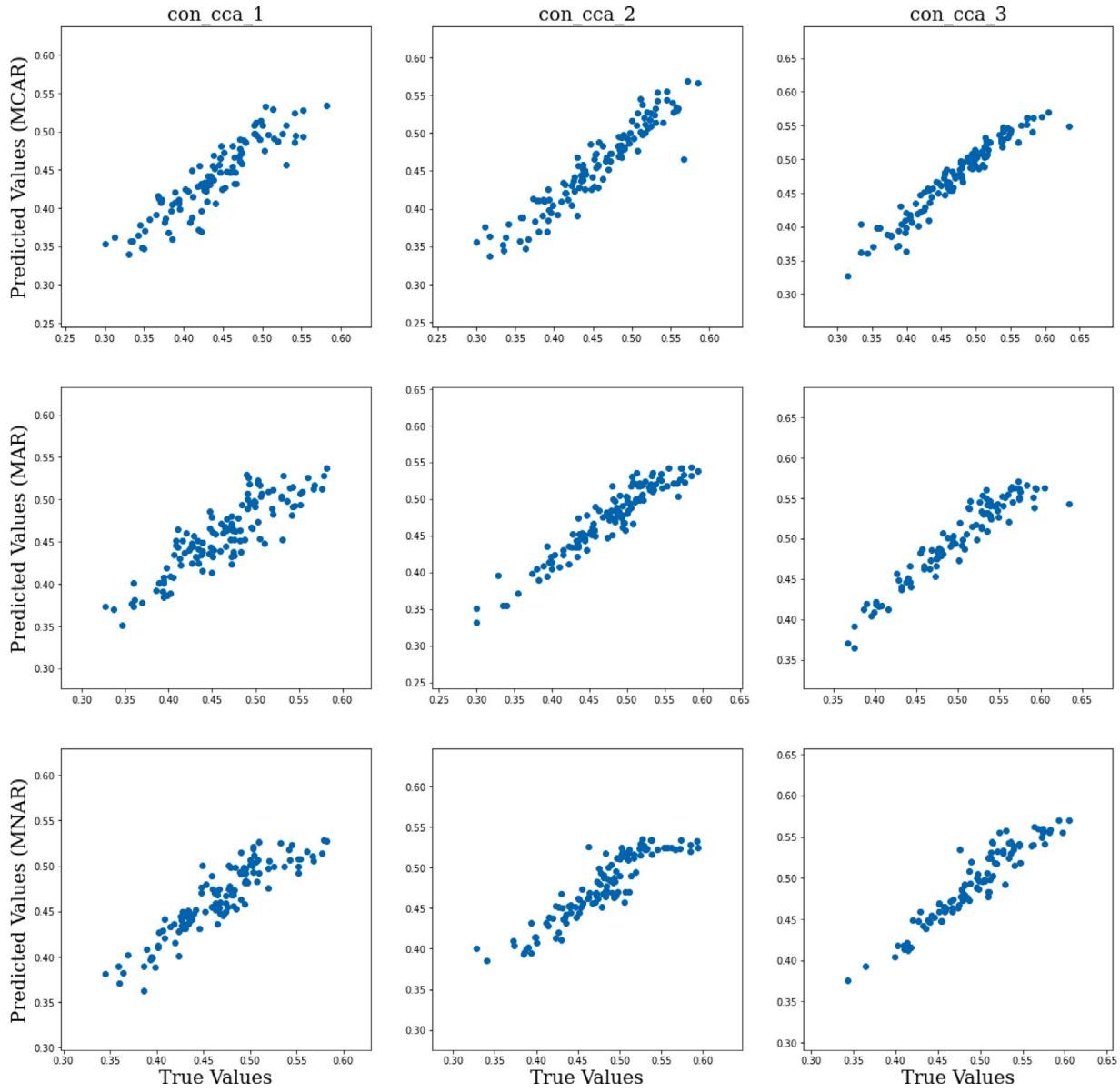


Fig. 12. The scatter plots of true v.s. predicted values of features in the study of DTI data (the case of 30% missing ratio) for MissForest.

In this DTI data example, we note that the scatter plots based on the predicted values using VAE tend to be horizontal. Those predicted values are randomly distributed around the median of the feature. They indicate that VAE resulted in the model collapse problem and generated bad results under all missing mechanism situations. Although GAIN was better than VAE, it seems that it is sensitive with the types of the missingness mechanisms. GAIN appears reasonable with data are MCAR, but becomes poor in cases where data are MAR or MNAR.

Deep generative modeling is an active research area. Many novel deep models with real applications have been proposed in recent years. The amount of research papers on deep learning for missing data imputation is growing and these novel methods appear potentially attractive. However, these models' power and validity need to be evaluated carefully in real applications. In many cases, reliable and reproducible code is either unavailable or incomplete. For the deep generative models, the number of hyperparameters to tune is usually much larger than in traditional statistical methods. The training time or memory size required for the hyperparameter search on big data can be prohibitive in some applications. Recently, Dong et al. (2021) have shown that GAIN had better accuracy than MICE and missForest in very large data. However, our findings demonstrated that the stability and convergence of the deep generative models were questionable for data with a small, moderate or even large sample size (*i.e.* when the sample size is less than 30,000).

Conventional methods have been well established in the scientific community, leading practitioners to use stable libraries instead of implementing deep learning alternatives. For example, the `mice` and `missForest` packages in R are well-tested and debugged packages for missing data imputation analysis. We have shown that both MICE and missForest methods are better and more stable methods for data with limited sample size. In comparison between MICE and missForest, the computational cost of missForest is much lower than MICE. Applying missForest does not require the standardization of the data, laborious dummy coding of categorical variables, and has no need for tuning parameters. MissForest can be applied to high-dimensional datasets where the number of variables greatly exceeds the sample size and still provides excellent imputation results (Stekhoven & Bühlmann, 2012). On the other hand, the advantages of MICE include that it results in unbiased estimates; its results are readily interpreted in a Bayesian context; many feasible algorithms are available within the MICE framework (Buuren & Groothuis-Oudshoorn, 2011). MICE is particularly

Table 2
Advantages and disadvantages of popular missing data imputation methods.

Methods	Advantages	Disadvantages
Statistical		
Mean/Median/Mode	• Simple and easy to implement • Computationally fast	• Might lead to severely biased estimates even if data are MCAR • Susceptible to skewed distributions
Regression	• Improvement over Mean/Median/Mode imputation	• The assumptions of error distribution and linear relationship are relatively strict • Poor results for heteroscedastic data
EM	• Performance equal to MICE when data are multivariate normal	• It has slow convergence
	• Guarantee that the likelihood will enhance after each iteration	• Underestimate standard error
MICE	• Widely used and accepted stable method • Flexibility	• Specification of conditional models which may be difficult to know a priori • Implementing MICE when data are MNAR may result in biased estimates
Machine learning		
K-Nearest Neighbors	• Easily handle both quantitative features and qualitative features • Accommodate missingness patterns	• Computational intensive for large data since it searches through all the dataset • Requires specification of hyperparameters that can have a large effect on the results
Feed-forward NN	• Good performance for skewed data	• Weight initialization is critical • Many models have to be constructed in a high-dimensional problem
MissForest	• Excellent performance for data with complex interactions and/or non-linear relations • Good for data with both quantitative features and qualitative features	• Training on observed data can be biased
Deep learning		
GAIN	• Data-driven approach and no distribution assumptions are needed	• Performance may be degraded depending on the sample size of data
	• Good for imbalanced and skewed data	• No standard software • Sensitive to the type of missing mechanism
VAE	• Can learn latent associations of the input data	• Implementation varies • Model collapse issues
DLIN	• Not sensitive to the missing mechanism • Can handle spatial or temporal data	• No standard software • Users need the advanced knowledge for model training

useful if missing values are associated with the target variable in a way that introduces leakage. MICE also allows users to make statements about the likely distribution of the missing value.

The effectiveness of different missing data imputation techniques depends on multiple factors, such as the sample size of the data, the distribution of the variables, the amount of missing values in the data, the correlation structure of the data, and the possible missingness mechanisms. In Section 2, we have summarized the popular missing data imputation methods. Here we address their advantages and disadvantages in Table 2. Practitioners should try to gather as much information as possible about the factors of missingness in a study, so that a suitable imputation technique could be appropriately applied.

6. Conclusion

In conclusion, we recommend that, for tabular data with a limited sample size ($n < 30,000$), practitioners use the conventional methods, MICE and missForest, to deal with missing data imputation in real case analyses. The recently developed deep learning method, DLIN, appeared to be a good alternative. The authors showed that it is not sensitive to the missing mechanism and the levels of missing ratios and can handle spatial or temporal data. However, we did not compare this method with others in our numerical studies because there is no publicly available open-source software for DLIN. Missing data imputation using deep learning techniques remains an active research area. There is great potential that deep learning-based methods will become popular in practice in the future, once reliable and easy-to-use software is developed.

In many situations, missing data imputation is just the first step of data analysis. Investigators will need to further build up casual regression models or risk prediction models based on the imputed datasets. Using appropriate imputation techniques is crucial to the success of those models. Successfully imputing missing predictor values can substantially increase the reliability of the risk predictions from the fitted model. They also need a good understanding of the dangers and limitations of the imputation technique they apply. A consent reporting guideline on how to report missing data imputation details in research studies is needed in future literature.

CRediT authorship contribution statement

Yige Sun: Software, Writing – original draft, Data curation. **Jing Li:** Writing – review & editing, Supervision. **Yifan Xu:** Writing – review & editing, Validation. **Tingting Zhang:** Writing – review & editing, Investigation. **Xiaofeng Wang:** Conceptualization, Methodology, Writing – original draft, Supervision.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

Acknowledgments

We are grateful to the Editor, the Associate Editor, and the reviewers for their constructive comments which substantially improved this paper. JL was supported in part by National Science Foundation (NSF), USA CCF-2006780, IIS-2027667, CCF-2200255 and National Institutes of Health (NIH), USA HG009658, 5U01AG073323.

Appendix A. Supplementary data

Supplementary material related to this article can be found online at https://doi.org/10.1016/j.eswa.2023.120201.

References

Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., et al. (2015). TensorFlow: Large-scale machine learning on heterogeneous systems. URL https://www.tensorflow.org/. Software available from tensorflow.org.

Agresti, A. (2003). *Categorical data analysis*. John Wiley & Sons, New York, NY.

Batista, G. E., & Monard, M. C. (2002). A study of K-nearest neighbour as an imputation method. *HIS - Frontiers in Artificial Intelligence and Applications*, *87*(87), 251–260.

Buuren, S. v., & Groothuis-Oudshoorn, K. (2011). Mice: Multivariate imputation by chained equations inR. *Journal of Statistical Software*, *45*, 1–67.

Cameron, A. C., & Trivedi, P. K. (2005). *Microeconometrics: Methods and applications*. Cambridge University Press.

Camino, R. D., Hammerschmidt, C. A., & State, R. (2019). Improving missing data imputation with deep generative models. (pp. 1–8). arXiv preprint arXiv:1902.10666.

Cannon, A. R., Cobb, G. W., Hartlaub, B. A., Legler, J. M., Lock, R. H., Moore, T. L., et al. (2018). *STAT2: Modeling with regression and ANOVA*. W.H. Freeman, New York, NY.

Chollet, F., et al. (2015). Keras. https://keras.io.

Dempster, A. P., Laird, N. M., & Rubin, D. B. (1977). Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society. Series B. Statistical Methodology*, *39*(1), 1–22.

Dong, W., Fong, D. Y. T., Yoon, J.-s., Wan, E. Y. F., Bedford, L. E., Tang, E. H. M., et al. (2021). Generative adversarial networks for imputing missing data for big data clinical research. *BMC Medical Research Methodology*, *21*(1), 1–10.

Erhan, L., Di Mauro, M., Anjum, A., Bagdasar, O., Song, W., & Liotta, A. (2021). Embedded data imputation for environmental intelligent sensing: A case study. *Sensors*, *21*(23), 7774.

Goldsmith, J., Crainiceanu, C. M., Caffo, B., & Reich, D. (2012). Longitudinal penalized functional regression for cognitive outcomes on neuronal tract measurements. *Journal of the Royal Statistical Society. Series C. Applied Statistics*, *61*(3), 453–469.

Gondara, L., & Wang, K. (2018). Mida: Multiple imputation using denoising autoencoders. In *Pacific-Asia conference on knowledge discovery and data mining* (pp. 260–272). Springer.

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., et al. (2020). Generative adversarial networks. *Communications of the ACM*, *63*(11), 139–144.

Gupta, A., & Lam, M. S. (1996). Estimating missing values using neural networks. *Journal of the Operational Research Society*, *47*(2), 229–238.

Hallaji, E., Razavi-Far, R., & Saif, M. (2021). DLIN: Deep ladder imputation network. *IEEE Transactions on Cybernetics*, *52*(9), 8629–8641.

Hastie, T., Tibshirani, R., Friedman, J. H., & Friedman, J. H. (2009). *The elements of statistical learning: Data mining, inference, and prediction* (2nd ed.). New York, NY: Springer.

Kingma, D. P., & Welling, M. (2014). Auto-encoding variational Bayes. arXiv:1312.6114.

Little, R. J., & Rubin, D. B. (2019). *Statistical analysis with missing data, vol. 793* (3rd ed). John Wiley & Sons.

Lu, H.-m., Perrone, G., & Unpingco, J. (2020). Multiple imputation with denoising autoencoder using metamorphic truth and imputation feedback. arXiv preprint arXiv:2002.08338.

McCoy, J. T., Kroon, S., & Auret, L. (2018). Variational autoencoders for missing data imputation with application to a simulated milling circuit. *IFAC-PapersOnLine*, *51*(21), 141–146, 5th IFAC Workshop on Mining, Mineral and Metal Processing MMM 2018.

Noordewier, M. O., Towell, G. G., & Shavlik, J. W. (1991). Training knowledge-based neural networks to recognize genes in DNA sequences. In *Advances in neural information processing systems* (pp. 530–536).

Qiu, Y. L., Zheng, H., & Gevaert, O. (2020). Genomic data imputation with variational auto-encoders. *GigaScience*, *9*(8), giaa082.

Ramsay, J., & Silverman, B. (2005). *Springer Series in Statistics*, *Functional data analysis* (2nd ed.). New York, NY: Springer.

Rasmus, A., Berglund, M., Honkala, M., Valpola, H., & Raiko, T. (2015). Semi-supervised learning with ladder networks. *Advances in Neural Information Processing Systems*, *28*.

Ripley, B. D. (2007). *Pattern recognition and neural networks*. Cambridge University Press.

Rubin, D. B. (1976). Inference and missing data. *Biometrika*, *63*(3), 581–592.

Rubin, D. B. (1978). Multiple imputations in sample surveys-a phenomenological Bayesian approach to nonresponse. In *Proceedings of the survey research methods section of the American statistical association, vol. 1* (pp. 20–34). VA, USA: American Statistical Association Alexandria.

Rubin, D. B. (2004). *Multiple imputation for nonresponse in surveys*. New York, NY: John Wiley & Sons.

Shah, A. D., Bartlett, J. W., Carpenter, J., Nicholas, O., & Hemingway, H. (2014). Comparison of random forest and parametric imputation models for imputing missing data using MICE: A CALIBER study. *American Journal of Epidemiology*, *179*(6), 764–774.

Sharpe, P. K., & Solly, R. (1995). Dealing with missing values in neural network-based diagnostic systems. *Neural Computing & Applications*, *3*(2), 73–77.

Stekhoven, D. J., & Bühlmann, P. (2012). MissForest—non-parametric missing value imputation for mixed-type data. *Bioinformatics*, *28*(1), 112–118.

Vincent, P., Larochelle, H., Bengio, Y., & Manzagol, P.-A. (2008). Extracting and composing robust features with denoising autoencoders. In *Proceedings of the 25th international conference on machine learning* (pp. 1096–1103).

Wahba, G., Gu, C., Wang, Y., & Chappell, R. (2018). Soft classification, aka risk estimation, via penalized log likelihood and smoothing spline analysis of variance. In *The mathematics of generalization* (pp. 331–359). CRC Press.

Waljee, A. K., Mukherjee, A., Singal, A. G., Zhang, Y., Warren, J., Balis, U., et al. (2013). Comparison of imputation methods for missing laboratory data in medicine. *BMJ Open*, *3*(8), Article e002847.

Yoon, J., Jordon, J., & Schaar, M. (2018). Gain: Missing data imputation using generative adversarial nets. In *International conference on machine learning* (pp. 5689–5698). PMLR.