Faster Imputation Using Singular Value Decomposition for Sparse Data

Phuc Nguyen[2,3], Linh G. H. Tran[2,3], Bao H. Le[1,2,3], Thuong H. T. Nguyen[2,3], Thu Nguyen[4], Hien D. Nguyen[3,5], and Binh T. Nguyen[1,2,3(✉)]

[1] AISIA Research Lab, Ho Chi Minh City, Vietnam
ngtbinh@hcmus.edu.vn
[2] University of Science, Ho Chi Minh City, Vietnam
[3] Vietnam National University, Ho Chi Minh City, Vietnam
[4] Simula Metropolitan, Oslo, Norway
[5] University of Information Technology, Ho Chi Minh City, Vietnam

Abstract. With the emergence of many knowledge-based systems worldwide, there have been more and more applications using different kinds of data and solving significant daily problems. Among that, the issues of missing data in such systems have become more popular, especially in data-driven areas. Other research on the imputation problem has dealt with partial and missing data. This study aims to investigate the imputation techniques for sparse data using the Singular Value Decomposition technique, namely SVDI. We explore the application of the SVDI framework for image classification and text classification tasks that involve sparse data. The experimental results show that the proposed SVDI method improves the speed and accuracy of the imputation process when compared to the PCAI method. We aim to publish our codes related to the SVDI later for the relevant research community.

Keywords: Sparse data · Data imputation · Singular Value decomposition

1 Introduction

The issue of missing data is a significant one that regularly emerges in many data-driven areas. Partial or missing data can occur due to several circumstances, including data entry mistakes, measurement flaws, or simply the inability to obtain specific information. It can result in skewed or incomplete studies and other problems, such as diminished statistical power, increased uncertainty, and poor interpretability. Imputation procedures, which fill in missing values in a data set to generate a complete data matrix, are frequently used to overcome this problem. On the other hand, sparse data refers to rows of data that include a significant percentage of zeroes as values. For example, it is frequently the case

Supported by Vietnam National University Ho Chi Minh City under the grant number DS2023-18-01.

© The Author(s), under exclusive license to Springer Nature Singapore Pte Ltd. 2023
N. T. Nguyen et al. (Eds.): ACIIDS 2023, LNAI 13995, pp. 135–146, 2023.
https://doi.org/10.1007/978-981-99-5834-4_11

in some issue areas, such as recommender systems, when a user has 0 ratings for all but a small number of movies or music in the database. Another typical illustration is a "bag of words" model of a text document, where most words have a value of 0, and other words in the document have a count or frequency. Examples of sparse data suitable for dimensionality reduction using Singular value decomposition (SVD) include Text Classification, One Hot Encoding, Bag of Words Counts, Recommender Systems, Customer-Product Purchases, User-Song Listen Counts, and User-Movie Ratings.

Singular value decomposition (SVD) is a widely used tool for data analysis with applications throughout science and engineering. In general, SVD operates by disassembling the initial matrix. SVD aims to approximate a dataset with many dimensions using fewer dimensions. The data are arranged in decreasing order of variation upon exposure of the substructure. This makes it easier to identify the area with the most variance, which may be reduced using SVD. By extracting the initial few singular vectors or eigenvectors, it may be used for dimension reduction, data visualization, data compression, and information extraction; for examples, see Alter et al. [1], Prasantha et al. [29], and Nguyen et al. [23,24]. On the one hand, SVD can solve several fundamental data analysis methods, such as the Principal component analysis (PCA) [8], the Canonical correlation analysis (CCA) [29], and the Singular Value Thresholding (SVT) [18]. On the other hand, SVD is also connected to several potent tools in different fields, such as the Latent matrix factorization (LMF) [33] and the Latent semantic analysis (LSA) [3]. Thus, when data is sparse, SVD may be the method with the highest level of popularity for dimensionality reduction.

In summary, the contribution of this paper can be listed as follows:

(a) We focus on evaluating the effectiveness of SVDI in parsing sparse data in two specific application domains: image and text classification.
(b) We compare the performance of SVD Imputation with Principal Component Analysis Imputation (PCAI) measures regarding their ability to handle missing data in these two application domains.

The rest of the paper is structured as follows. Section 2 presents a survey about imputation methods and their application in practical problems. Section 3 describes the process combining Singular Value Decomposition and imputation techniques. The results and discussion are performed in Sect. 4. The paper ends with conclusions and future works in the last section.

2 Related Works

Instead of removing or ignoring the unknown data, a large amount of research has been tackling this problem by using imputation methods [25]. While Suthar et al. [30] surveys classifying the imputation methods of missing data in data mining, Musil et al. [22] and Lüdtke et al. [19] compare imputation strategies

in different designs. There are also advanced imputation methods, such as K-nearest Neighbor (KNN) imputation [20] and Machine Learning-based imputation [10,12]. Besides that, Lakshminarayan et al. [12] experiment with two Machine Learning (ML) systems: Autoclass and C4.5, for the problem.

In addition to various methods available for handling missing data, one noteworthy example is the Generative Adversarial Imputation Nets (GAIN) [34], which is a modified version of the Generative Adversarial Nets (GAN) framework. GAIN uses a generator and discriminator network to impute missing data and is trained using additional information as a hint vector to focus on imputation quality. Another technique is the Missing GP (MGP) [9], which uses sparse Gaussian processes to predict missing values at each dimension using all the variables from other dimensions. MGP outputs a predictive distribution for each missing value and can be trained simultaneously to impute all observed missing values. Finally, Khan et al. [11] suggests a hybrid technique of single and multiple imputation techniques, which extends the Multivariate Imputation by Chained Equation (MICE) algorithm to impute categorical and numeric data. Additionally, Awan et al. [2] presents the Conditional Generative Adversarial Imputation Network (CGAIN), which imputes missing data using class-specific distributions based on class-specific characteristics of the data. Moreover, DPER algorithm [26] directly computes maximum likelihood estimates (MLEs) for randomly missing data sets, eliminating the need for separate imputation steps. It provides computational efficiency and superior estimation performance compared to existing methods.

Many studies have applied to impute methods to address missing data in real-world problems. Firstly, Jerez et al. [10] use statistical and machine learning methods to impute missing data in an actual breast cancer problem. Furthermore, Liu et al. [16] have a systematic review of deep learning-based imputation techniques for handling missing values in healthcare data. The study aims to evaluate the use of these techniques, with a particular focus on data types, to assist healthcare researchers in dealing with missing values. Hassan et al. [7] propose a missing data imputation method based on the salp swarm algorithm for diabetes disease. The study aims to impute missing values in the Pima Indian diabetes disease dataset using a proposed algorithm, namely ISSA.

Singular value decomposition (SVD) has recently been widely used in different fields, including multi-environment trials and transforming genome-wide expression data. However, in the research of Alter et al., [1], imputing missing values using standard SVD can lead to low-quality results when affected by outliers. Still, the Yan method proposed four robust SVD extensions to address this issue. Singular value decomposition can also be used in transforming genome-wide expression data [5], enabling meaningful comparisons of the expression of different genes across different arrays and experiments. Moreover, to improve the issue of handling missing data, a proposed Bayesian model that is based on the SVD components of a continuous data matrix is shown by Zhai et al. [35] to be the most accurate and precise method compared to the current imputation methods in simulated and real datasets.

3 Methodology

3.1 Sparse Data

Sparse data refers to datasets characterized by a significant proportion of zero or missing entries, indicating that the vast majority of the data points possess values of zero. Unlike missing data, where values are unknown or undefined, sparse data values are generally known but non-existent or specifically set to zero. This terminology finds frequent application in fields such as machine learning, data science, and information retrieval, where dealing with sparse data poses a common challenge.

Consider a movie recommendation system that suggests movies to users based on their viewing history. The system has a comprehensive database with information about movies, users, and ratings. However, not all users have watched or rated every movie, resulting in a sparse dataset with many missing entries. For example, in a dataset with 100,000 users and 1,000 movies, only 1 million non-zero entries exist, representing less than 1% of the total dataset. This high sparsity means that most entries in the dataset are zero values.

3.2 Mechanisms of Missing Data

The impact of missing data depends on the method used to generate the missing data. Rubin and his colleagues [13–15,28] established the foundations of missing data theory. Central to missing data theory is his classification of missing data problems into three categories: (1) missing completely at random (MCAR); (2) missing at random (MAR); and (3) missing not at random (MNAR). These three classes of missing data are referred to as missing data mechanisms (for a slightly different classification, see [6]). Despite the name, they are not reasons for missing data that are causative. Instead, the statistical link between observations (variables) and the risk of missing data is represented by missing data mechanisms. Another word that is sometimes confused with missing data mechanisms is missing data patterns; these are descriptions of which values in a dataset are missing.

MCAR occurs when the missing data is independent of the observed or unobserved data. For example, participants flip a coin in a survey to decide whether to answer questions. MAR occurs when the missingness can be explained using observed data. For instance, survey participants that live in specific postal codes may refuse to fill in the questionnaire. MNAR occurs when the missingness depends on an unobserved or missing attribute. For example, people who own six-bedroom houses may refuse to participate in a survey, as owning a bigger house may indicate greater wealth and a better-paying job. A researcher's choice of approach is made more accessible when the data are MCAR or MAR since they allow them to overlook the causes of missing data. Every approach is viable in this situation. The data may be MCAR or MAR, but it is challenging to provide actual proof of this. It is a sound method to compare findings from many studies to see their sensitivity to the MCAR and MAR assumptions. The outcomes of the various analyses differ from one another, and this reveals which assumptions are the most important.

3.3 Singular Value Decomposition (SVD)

The concept of singular value decomposition (SVD) is a fundamental tool in linear algebra. Given an $n \times d$ matrix A, one can express it as the product of three matrices:

$$A = USV^T, \tag{1}$$

where U is an $n \times n$ orthogonal matrix, V is a $d \times d$ orthogonal matrix, and S is an $n \times d$ diagonal matrix with nonnegative entries. Notably, the diagonal entries of S are sorted from highest to lowest, progressing from the "northwest" to the "southeast" of the matrix. Assuming we use r eigenvalues, the projection matrix can be defined as $V = W_r$, where W_r is formed by selecting the first r columns of matrix V. Consequently, the reduced dimension version of matrix A is given by the product AV as well.

The following formula, as depicted below, is helpful in providing a more detailed illustration of the singular value decomposition (SVD) method. SVD is a powerful mathematical tool used for matrix factorization, wherein each singular value in S is accompanied by an associated left singular vector in U and a right singular vector in V.

$$\mathrm{A} = \mathrm{USV^T} = \underset{n\times n}{\begin{bmatrix} | & | & \& | \\ u_1 & u_2 & \dots\ u_n \\ | & | & \& | \end{bmatrix}} \times \underset{n\times d}{\begin{bmatrix} \sigma_1 & 0 & \dots & 0 \\ 0 & \sigma_2 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \sigma_d \\ 0 & 0 & \vdots & 0 \end{bmatrix}} \times \underset{d\times d}{\begin{bmatrix} | & v_1^T & | \\ | & v_2^T & | \\ \& & \vdots & \\ | & v_d^T & | \end{bmatrix}}$$

It is important to note that the orthogonal matrices U and V that are part of the singular value decomposition (SVD) of matrix A are not necessarily the same. It is because A may not be a square matrix, resulting in U and V having different dimensions. The columns of U represent the left singular vectors of A, while the columns of V, or the rows of V^T, represent the right singular vectors of A. The singular values of matrix A are represented by the entries of S, with each singular value being associated with a singular vector. The singular vectors are ordered such that the first or top singular vector corresponds to the largest singular value, which is illustrated in the figure above. It is worth noting that every matrix A can be decomposed into its SVD, a remarkable fact with a straightforward proof that is better suited for a linear algebra course. Geometrically, this means that no matter how peculiar a matrix may be, it can always be decomposed into a rotation (multiplication by V^T), scaling plus dimension addition or removal (multiplication by S), and a rotation within the range (multiplication by U). The SVD is "more or less unique," with the singular values of a matrix being unique. If a singular value is repeated, the subspaces created by the corresponding left and right singular vectors have a distinct definition. However, there is flexibility in selecting orthonormal bases for each of these subspaces.

3.4 SVD Imputation (SVDI)

The below algorithms describe the approach of "SVD Imputation" (SVDI). For example, suppose there exists a dataset $\mathcal{D} = [\mathcal{F}, \mathcal{M}]$, which can be decomposed into a partition of fully observed features denoted by $\mathcal{F}$, and another partition $\mathcal{M}$ containing features with missing values. To facilitate the imputation process on this incomplete dataset, one may adopt the approach of SVD Imputation (SVDI), which involves reducing the dimensionality of the fully observed partition $\mathcal{F}$ via $svd(A)$, generating a new reduced feature matrix $\mathcal{R}_{\mathcal{F}}$. Subsequently, the imputation process can be carried out on the union of $\mathcal{R}_{\mathcal{F}}$ and the partition with missing values, $\mathcal{M}$, instead of the original full dataset $[\mathcal{F}, \mathcal{M}]$.

The rationale for this approach is twofold. First, by reducing the dimensionality of $\mathcal{F}$, one can accelerate the computational efficiency of the imputation method. This is particularly beneficial in scenarios where the size of the covariance matrix, a key component of SVD, is smaller than that of the full dataset $\mathcal{F}$ due to $\mathcal{F}$ having more samples than features. In such cases, implementing SVD based on the covariance matrix is expected to be faster. Conversely, when the number of features in $\mathcal{F}$ exceeds the sample size, the covariance matrix of $\mathcal{F}$ is larger than that of $\mathcal{F}$ itself. In this scenario, SVD formulation based on the data is a more favorable approach. By considering these factors, researchers and practitioners can optimize the SVDI methodology to suit their particular dataset and computational resources best. The variations in the mean squared error of the imputed version and the ground truth for various procedures are only marginally different, as demonstrated in the studies. SVDI appears to perform somewhat better on several occasions. That is feasible because SVD keeps just the essential information from the data while eliminating some noise, improving imputation quality.

Algorithm 1. SVD imputation framework

Require:
$\mathcal{D} \leftarrow [\mathcal{F}, \mathcal{M}]$
Imputer I
SVD algorithm svd
Procedure:
$(\mathcal{R}, V) \leftarrow svd(\mathcal{F})$
$\mathcal{M}' \leftarrow I([\mathcal{R}, \mathcal{M}])$
Return Imputed version $\mathcal{M}'$ of $\mathcal{M}$

4 Experiments

In this section, we validate the performance of SVDI using multiple real-world datasets with various settings (such as on datasets with different missing rates), and we compare SVDI and PCAI when the objective is to perform classification on the imputed dataset. We report RMSE, running time, and average accuracy

as the standard performance metric. Unless specified, missingness is applied to the datasets with a missing rate is 20%, and the default missing mechanism is MCAR.

4.1 Datasets

We experiment on three datasets:

1. **IMDB**[1] The IMDB dataset contains 50000 movies and TV shows divided into 25000 training and 25000 test samples. Each review is labeled as positive or negative based on sentiment. The reviews are preprocessed. Each review sentence can be represented by TF-IDF 5000 features.
2. **Fashion MNIST**[2] includes clothing images is also selected in our experiments. The dataset consists of 60000 training images, 10000 testing images of size 28 $\times$28 (784 features), and ten labels corresponding to 10 different types of fashion.
3. **MNIST**[3] is a large collection of handwritten digits. It also has a training set of 60000 images, a test set of 10000 images of size 28$\times$28, and ten labels corresponding to 10 digits.

The detail of each dataset can be listed below (TAble 1).

Table 1. The description of datasets used in our experiments.

Dataset	# classes	# features	Samples	Sparsity
Fashion MNIST	10	784	70000	50.1
MNIST	10	784	70000	80.4
IMDB	2	5000	50000	98.05

4.2 Experimental Design

We compare the running time, average RMSE, and average accuracy of PCAI with our SVD Imputation (SVDI) methods. We calculate the running time by the sum of dimensional reduction and imputation time. The imputation methods we use in our experiments are:

1. **SoftImpute** [21]: Matrix completion by iterative soft thresholding of SVD decompositions. The algorithm fills in the missing values with the current guess and then solves the optimization problem on the complete matrix using a soft-thresholded SVD.

[1] https://www.kaggle.com/datasets/lakshmi25npathi/imdb-dataset-of-50k-movie-reviews.
[2] https://github.com/zalandoresearch/fashion-mnist.
[3] http://yann.lecun.com/exdb/mnist/.

2. **Multiple Imputation by Chained Equation (MICE)** [4]: models each feature with missing values as a function of other features and uses that estimate for imputation in an iterated round-robin fashion.
3. **kNN Imputation (KNNI)** [32]: Nearest neighbor imputations which weight samples using the mean squared difference on features for which two rows both have observed data.
4. **GAIN** [34]: A deep learning approach for imputing missing data by utilizing Generative Adversarial Network (GAN).

All methods are implemented with default configurations in their original papers. For all PCA computations, the number of eigenvectors is chosen so that the minimum amount of variance explained is 95%. We utilize logistic regression as the classifier.

It is worth noting that any dataset can be rearranged so that the first q features are not missing while the remaining features have missing values. Therefore, we assume that each dataset's first q features are not missing, while the remaining ones contain missing values. The default value for q is half of the total number of attributes in each dataset. Then, we randomly simulate missing data in the missing partition M at default missing rates of 20%, 40%, and 60%. Here, a missing rate of x% refers to the percentage of missing entries in the missing partition M. To introduce a fixed missing rate, we use two different missing mechanisms inspired by [17].

(a) **MCAR**: Set all features in missing partition M to have missing values when $v_i \leq, t, i \in (1:n)$ rate with t is the missing rate.
(b) **MNAR**: Randomly sample 2 features x_1 and x_2 from the missing partition M, calculate their median m_1 and m_2. Then we set all features to the missing value where $v_i \leq t, i \in (1:n)$ and ($x_1 \leq m_1$ or $x_2 \leq m_2$) and t is the missing rate.

Unless otherwise stated, missingness is applied to the datasets by randomly removing 20% of all missing partition M, with MCAR as the default missing mechanism. We conduct all experiments using the Kaggle notebook with a default Intel Xeon CPU and 30 GB RAM. If no results are produced after 20000 s of running, or if a memory allocation issue arises, we terminate the experiment and denote it as **NA** in the result tables.

4.3 Results and Discussion

We perform experiments comparing the performance between SVDI and PCAI and present the results in the tables below. From now on, the bold values on the tables indicate better performance for each metric on each dataset. According to Table 2, SVDI outperforms PCAI by having lower average RMSE values for most methods and datasets. Specifically, SVDI proves more effective than PCAI with SoftImpute, Mice, and KNNI methods on Fashion MNIST and MNIST datasets. In the IMDB datasets, there is an insignificant difference between PCA and SVD, and we could not retrieve the results of MICE due to a memory issue.

Table 2. The average RMSE of Imputation methods.

Methods	Strategy	Fashion MNIST	MNIST	IMDB
SoftImpute	PCAI	0.20675	0.3618	**0.448**
	SVDI	**0.155**	**0.1995**	0.458
GAIN	PCAI	**0.156**	1.2557	0.452
	SVDI	0.261	**0.1754**	**0.443**
Mice	PCAI	0.244	2.712	**NA**
	SVDI	**0.09**	**0.11**	
KNNI	PCAI	0.162	1.2495	**0.6204**
	SVDI	**0.122**	**0.166**	0.6211

Table 3. The average accuracy (%) of Imputation methods.

Methods	Strategy	Fashion MNIST	MNIST	IMDB
SoftImpute	PCAI	**84.19**	92.05	85.94
	SVDI	83.64	**92.38**	**88.51**
GAIN	PCAI	**84.06**	**92.5**	85.952
	SVDI	83.9	92.3	**88.548**
Mice	PCAI	80.1	92.43	**NA**
	SVDI	**83.8**	**92.46**	
KNNI	PCAI	**84.35**	**92.85**	85.9
	SVDI	83.84	92.83	**88.54**

But overall, SVDI is a better strategy for imputing missing values than PCAI, especially when combined with SoftImpute, Mice, and KNNI methods.

Based on the results presented in Table 3, the average accuracy table, it is evident that the SVDI imputation strategy generally surpasses the PCAI strategy. GAIN and KNNI with PCAI achieve slightly higher average accuracy on the Fashion and MNIST datasets than their SVDI counterparts. However, on the IMDB dataset, PCAI performs notably worse, with an average accuracy of nearly 3% lower than that of SVDI. Mice method with PCAI tends to yield lower average accuracies on the Fashion MNIST and MNIST datasets, with values of 80.1% and 92.43%, respectively. For the SoftImpute technique, while PCAI outperforms SVDI on the Fashion MNIST, the MNIST dataset shows that PCAI is 0.33% less effective than SVDI. Notably, all SVDI strategies yielded results approximately 3% better than PCAI ones on the IMDB dataset.

Table 4 displays each method's average running time values on different datasets. PCAI demonstrates faster running time in the Fashion MNIST dataset with all four methods compared to SVDI. On the contrary, in the IMDB and MNIST datasets, SVDI is consistently more effective than PCAI, with lower running time values. Overall, it is evident that SVDI exhibits faster perfor-

Table 4. The average running time (s) of Imputation methods.

Methods	Strategy	Fashion MNIST	MNIST	IMDB
SoftImpute	PCAI	**17.92**	18.97	644.24
	SVDI	18.38	**17.81**	**411.47**
GAIN	PCAI	**383.9**	**417.63**	14070.26
	SVDI	422.9	423.58	**10813.95**
Mice	PCAI	**1984.63**	**4664.094**	**NA**
	SVDI	2664.87	6211.2	
KNNI	PCAI	**6376**	11314.2	18490.52
	SVDI	10835	**10156.6**	**17268.33**

mance than PCAI on the MNIST and IMDB datasets, which can be attributed to SVDI's omission of the standardization step, resulting in quicker processing time.

5 Conclusion and Future Works

This study explores the application of the SVDI framework for image classification and text classification tasks that involve sparse data. The experimental setup consists of the dimensionality reduction of fully observed features and subsequent imputation of missing data using the reduced feature set. This approach enables us to effectively handle the high-dimensional and sparse data common in image and text classification while also addressing the issue of missing data that frequently arises in real-world datasets. The average RMSE, average accuracy, and running time are used as evaluation metrics to compare the performance of SVDI with other dimension reduction imputation methods, such as PCAI. After conducting experiments, it illustrates that the SVDI method improves the speed and accuracy of the imputation process when compared to the PCAI method.

For future work, the proposed method can be experimented on visible and thermal infrared image datasets, such as KTFE [27] and USTC-NVIE [31]. The objective is to comprehensively investigate these frameworks to gain deeper insights into the underlying factors contributing to this discrepancy and develop viable solutions to address these challenges.

Acknowledgments. This research is funded by Vietnam National University Ho Chi Minh City in Vietnam under the funding/grant number DS2023-18-01.

References

1. Alter, O., Brown, P.: Processing and modeling genome-wide expression data using singular value decomposition. In: Proceedings of SPIE - The International Society for Optical Engineering, vol. 4266 (2001)

2. Awan, S.E., Bennamoun, M., Sohel, F., Sanfilippo, F., Dwivedi, G.: Imputation of missing data with class imbalance using conditional generative adversarial networks. Neurocomputing **453**, 164–171 (2021)
3. Berry, M., Dumais, S., Gavin, W.: O'brien, using linear algebra for intelligent information retrieval. SIAM Rev. **37**, 573–595 (1995)
4. van Buuren, S., Groothuis-Oudshoorn, K.: mice: multivariate imputation by chained equations in R. J. Stat. Softw. **45**(3), 1–67 (2011). https://doi.org/10.18637/jss.v045.i03. https://www.jstatsoft.org/index.php/jss/article/view/v045i03
5. García-Peña, M., Arciniegas-Alarcón, S., Krzanowski, W.J., Duarte, D.: Missing-value imputation using the robust singular-value decomposition: proposals and numerical evaluation. Crop Sci. **61**(5), 3288–3300 (2021)
6. Gelman, A., Hill, J.: Data analysis using regression and multilevel/hierarchical models (2007)
7. Hassan, G.S., Ali, N.J., Abdulsahib, A.K., Mohammed, F.J., Gheni, H.M.: A missing data imputation method based on salp swarm algorithm for diabetes disease. Bull. Electric. Eng. Inf. **12**(3), 1700–1710 (2023)
8. Huang, J., Shen, H., Buja, A.: The analysis of two-way functional data using two-way regularized singular value decompositions. J. Am. Stat. Assoc. **10**4, 1609–1620 (2009)
9. Jafrasteh, B., Hernández-Lobato, D., Lubián-López, S.P., Benavente-Fernández, I.: Gaussian processes for missing value imputation (2022)
10. Jerez, J.M., et al.: Missing data imputation using statistical and machine learning methods in a real breast cancer problem. Artif. Intell. Med. **50**(2), 105–115 (2010)
11. Khan, S.I., Hoque, A.S.M.L.: SICE: an improved missing data imputation technique. J. Big Data **7**(1), 1–21 (2020)
12. Lakshminarayan, K., Harp, S.A., Goldman, R.P., Samad, T., et al.: Imputation of missing data using machine learning techniques. In: KDD, vol. 96 (1996)
13. Little, R., Rubin, D.: Regression with missing XS - a review. J. Am. Stat. Assoc. **87**, 1227–1237 (1992)
14. Little, R., Rubin, D.: Modeling the drop-out mechanism in repeated-measures studies. J. Am. Stat. Assoc. **90**, 1112–1121 (1995)
15. Little, R., Rubin, D.: Statistical analysis with missing data (2014)
16. Liu, M., et al.: Handling missing values in healthcare data: a systematic review of deep learning-based imputation techniques. Artif. Intell. Med., 102587 (2023)
17. Gondara, L., Wang, K.: MIDA: multiple imputation using denoising autoencoders. In: Phung, D., Tseng, V.S., Webb, G.I., Ho, B., Ganji, M., Rashidi, L. (eds.) PAKDD 2018. LNCS (LNAI), vol. 10939, pp. 260–272. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-93040-4_21
18. Lu, C., Zhu, C., Xu, C., Yan, S., Lin, Z.: Generalized singular value thresholding. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 29 (2015)
19. Lüdtke, O., Robitzsch, A., Grund, S.: Multiple imputation of missing data in multilevel designs: a comparison of different strategies. Psychol. Methods **22**(1), 141 (2017)
20. Malarvizhi, R., Thanamani, A.S.: K-nearest neighbor in missing data imputation. Int. J. Eng. Res. Dev. **5**(1), 5–7 (2012)
21. Mazumder, R., Hastie, T., Tibshirani, R.: Spectral regularization algorithms for learning large incomplete matrices. J. Mach. Learn. Res. **11**(80), 2287–2322 (2010). http://jmlr.org/papers/v11/mazumder10a.html
22. Musil, C.M., Warner, C.B., Yobas, P.K., Jones, S.L.: A comparison of imputation techniques for handling missing data. West. J. Nurs. Res. **24**(7), 815–829 (2002)

23. Nguyen, H.D., Sakama, C., Sato, T., Inoue, K.: Computing logic programming semantics in linear algebra. In: Kaenampornpan, M., Malaka, R., Nguyen, D.D., Schwind, N. (eds.) MIWAI 2018. LNCS (LNAI), vol. 11248, pp. 32–48. Springer, Cham (2018). https://doi.org/10.1007/978-3-030-03014-8_3
24. Nguyen, H.D., Sakama, C., Sato, T., Inoue, K.: An efficient reasoning method on logic programming using partial evaluation in vector spaces. J. Log. Comput. **31**(5), 1298–1316 (2021)
25. Nguyen, T., Nguyen, D.H., Nguyen, H., Nguyen, B.T., Wade, B.A.: EPEM: efficient parameter estimation for multiple class monotone missing data. Inf. Sci. **567**, 1–22 (2021)
26. Nguyen, T., Nguyen-Duy, K.M., Nguyen, D.H.M., Nguyen, B.T., Wade, B.A.: DPER: direct parameter estimation for randomly missing data. Knowl.-Based Syst. **240**, 108082 (2022)
27. Nguyen, V., Tran, N., Nguyen, H., et al.: KTFEv2: multimodal facial emotion database and its analysis. IEEE Access **11**, 17811–17822 (2023)
28. Rubin, D.: Inference and missing data. Biometrika **63**, 5781–590 (1976)
29. Prasantha, H.S., Shashidhara, H.L., Murthy, K.B.: Image compression using SVD. In: International Conference on Computational Intelligence and Multimedia Applications, pp. 143–145 (2008)
30. Suthar, B., Patel, H., Goswami, A.: A survey: classification of imputation methods in data mining. Int. J. Emerg. Technol. Adv. Eng. **2**(1), 309–12 (2012)
31. Wang, S., Liu, Z., Lv, S., et al.: A natural visible and infrared facial expression database for expression recognition and emotion inference. IEEE Trans. Multimedia **12**(7), 682–691 (2010)
32. Woźnica, K., Biecek, P.: Does imputation matter? benchmark for predictive models. In: 37th International Conference on Machine Learning (2020)
33. Yang, D., Ma, Z., Buja, A.: A sparse SVD method for high-dimensional data. J. Comput. Graph. Stat. **23**, 923–942 (2014)
34. Yoon, J., Jordon, J., van der Schaar, M.: Gain: missing data imputation using generative adversarial nets (2018)
35. Zhai, R., Gutman, R.: A Bayesian singular value decomposition procedure for missing data imputation. J. Comput. Graph. Stat., 1–13 (2022)