Imputing missing values using cumulative linear regression

Samih M. Mostafa ✉

Faculty of Science, Mathematics Department, Computer Science, South Valley University, Qena 83523, Egypt
✉ *E-mail: samih_said@sci.svu.edu.eg*

Abstract: The concept of missing data is important to apply statistical methods on the dataset. Statisticians and researchers may end up to an inaccurate illation about the data if the missing data are not handled properly. Of late, Python and R provide diverse packages for handling missing data. In this study, an imputation algorithm, cumulative linear regression, is proposed. The proposed algorithm depends on the linear regression technique. It differs from the existing methods, in that it cumulates the imputed variables; those variables will be incorporated in the linear regression equation to filling in the missing values in the next incomplete variable. The author performed a comparative study of the proposed method and those packages. The performance was measured in terms of imputation time, root-mean-square error, mean absolute error, and coefficient of determination (R^2). On analysing on five datasets with different missing values generated from different mechanisms, it was observed that the performances vary depending on the size, missing percentage, and the missingness mechanism. The results showed that the performance of the proposed method is slightly better.

1 Introduction

Scientific studies of statistical learning use data-dependent tool (e.g. machine learning and the like) to find a predictive model based on the data. Undoubtedly, better data quality leads to a better model; therefore, better analysis and prediction. In real-world data, the dataset is likely to contain missing values, which occur when one or more variables contain no values in one or more observations. Missing data reduces the statistical performance and produces biased estimates of a study, leading to invalid results [1].

1.1 Motivation and novelty

Some existing imputation algorithms fail in imputation of the missing data; others take a long time for imputation or give poor performance. This paper treats these defects by proposing a novel imputation method that exploits the most influential variables. The priority of variables to be selected in the imputation depends on some criteria, which will be discussed in Section 3.

1.2 Contributions of this paper

The main contributions of this work are: it gives an outline of the studies related to dealing with missing data, shows the advantages and disadvantages of these studies, shows how the performance metrics affected by the size of the dataset, proposes an imputation approach which benefits from all the variables to improve the quality of data, and compares between the proposed method and most popular R and Python imputation methods in all missingness mechanisms for different datasets with different sizes.

1.3 Missingness mechanisms

Missing values are a common occurrence, and the conclusions drawn from the data are significantly affected by the missing data. Various reasons of the missingness, for example, not limited, are: individuals do not know the answer or refuse to answer, sensor failure, error in data transfer, errors when collecting the data, data entry error, and so on. The relationship between known variables and the probability of the missingness is defined as the *missingness mechanism* (i.e. why the data are missing) [2–11]. Missing values are categorised into three types:

- *Missing completely at random (MCAR):* If the probability of missingness is the same for all observations. The reason for the missingness in a variable X does not depend on any variables as well as X itself on the dataset.
- *Missing at random (MAR):* If the probability of missingness is the same only within the observed data. The reason for the missingness in a variable X depends on other variables on the data but does not depend on the variable X itself.
- *Missing not at random (MNAR):* If the probability of the missingness for a variable X depends on X itself or other variables not completely known.

1.4 Handling missing data

Data-dependent tools deal only with complete datasets (i.e. with no missing values). Therefore, it is vitally important to handle missing values. The two approaches for dealing with missing values are deletion and imputation [12].

1.4.1 Deletion approach:
Many missing data methods ease the problem with the disposal of the data. Complete case analysis, aka listwise, is a direct approach for handling missing values by excluding them from the dataset. Although it is easy to implement, most of the data would be discarded if many variables contain unknown values, this leads to reducing the sample size and may be very few complete cases exist; this may lead to an unpredictable bias and estimates with larger standard errors. To remedy the loss of the data, available case analysis, aka pairwise, is another approach that uses observations that contain missing values. However, statistical procedure cannot use the feature if it contains missing values; the case with missing values can still be used when analysing other features with recorded values. Pairwise outperforms listwise, where it uses more data. However, each

statistical analysis may be based on a different subset of the observations; this can be suspicious [13].

1.4.2 Imputation approach:
Instead of discarding incomplete data, replacing missing values by appropriate values using the information available to conjecture the value of missing value, called imputation, is an alternative approach for handling missing values. Using imputation, most, if not all, data will be used for statistical methods. Imputation technique should be selected carefully, where the performance of the imputation technique is affected by the richness of missingness and the missingness mechanism, thence, the selection of imputation technique affects the quality of the data. Depending on the imputation mechanism, imputation techniques can be classified into two groups: intransitive, in which the imputation of a variable of interest, which contains missing values, depends on itself, not other variables, and transitive, in which the imputation of a variable of interest depends on other variables. The arithmetic mean, aka unconditional means, mode, median, and most frequently are examples of intransitive imputation. Interpolation and regression are examples of transitive imputation. Imputation can be done in two ways: single imputation and multiple imputation. In the former, each missing value is imputed by one plausible value [14]. In the latter, proposed by Rubin [15], each missing value is imputed n times, $n > 1$, which generates n versions of the complete datasets [16]. Imputation can also be estimated using regression methods [17], K-nearest neighbours (KNNs), hot-deck imputation [18] etc. As the proposed algorithm depends on the linear regression, the following are the most common regressions used:

- *Simple linear regression:* If there is a linear relationship between the dependent variable y and the independent variable X, the mathematical notation can be written as below:

$$y = \beta_0 + \beta_1 X + \varepsilon \qquad (1)$$

Equation (1) says: Regressing the dependent y on the independent X. Value of unknown y based on $X \mid \forall X = x$ can be predicted by

$$\hat{y} = \hat{\beta}_0 + \hat{\beta}_1 X \qquad (2)$$

The hat symbol $\hat{}$ indicates the estimated value of the unknown coefficient/parameter:
- *Multiple linear regression:* More independent variables work together to achieve better prediction. The linear relationship between the dependent and independent variables can be written as below:

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_p x_p + \varepsilon \qquad (3)$$

1.5 Organisation

The rest of this paper is organised as follows: Section 2 presents the literature review; the proposed method is discussed in Section 3; Section 4 discusses the experimental implementation; and conclusion, findings, and future work are briefly explained in Section 5.

2 Literature review

This section presents an overview of the studies related to dealing with missing values.

Cismondi *et al.* [2] presented a method for handling missing data in intensive care units databases to improve the modelling performance. The authors determined which missing data should be imputed by using statistical classifier followed by fuzzy modelling. The authors developed a simulation test bed to estimate the performance. Although the authors' approach improved the accuracy of classifications, sensitivity, and specificity, the approach may fail in filling all missing values.

Hapfelmeier *et al.* [3] formulated the imputation problem as an optimisation problem. Within their proposed framework, the authors used a support vector machine (SVM), KNNs, and decision tree. The framework incorporates two composite methods: opt.cv and benchmark.cv. The former selects the best approach from opt.svm, opt.knn, and opt.tree. The latter selects the best method from mean, predictive-mean matching, Bayesian principal component analysis, KNNs, and iterative KNNs. Although their proposed method outperforms other methods, the time used for selecting the best method, which gives the lowest mean absolute error (MAE) is long. Also, the sizes of datasets used in the experiments are small.

Batista and Monard [4] compared between KNN, C4.5, and CN2. The experiments were implemented at a different rate of missing values. Although the analysis indicates that the KNN method outperforms C4.5 and CN2 even when the dataset contains a high percentage of missingness, in some cases, C4.5 is competitive to ten-nearest neighbour. To confirm the analysis significance, the value of k should be increased.

Aydilek and Arslan [5] combined support vector regression and genetic algorithm (GA) with fuzzy clustering to impute missing data. Their proposed method was compared with FcmGa, Zeroimpute, and SvrGa methods. Although the imputation accuracy was better, the efficiency of the training stage by the support vector regression depends on the size of the complete dataset [i.e. in which no variables have missing value(s)]; this means that if many attributes contain many missing values, many cases will be discarded.

Qin *et al.* [19] proposed an imputation method called stochastic semi-parametric regression for semi-parametric data and compared with deterministic semi-parametric regression imputation. The authors aimed at making an optimal evaluation about root-mean-square error (RMSE), and evaluated their proposed method using real data and simulated data experimentally. Although their proposed approach is better than deterministic semi-parametric imputation in effectiveness and efficiency, the authors used two accuracy measures: mean squared error (MSE) and RMSE, both of them are susceptible to outliers since they give extra weight to large error [20].

Acuña and Rodriguez [21] compared four popular approaches: complete case analysis, mean imputation, median imputation, and KNN imputation (KNNI) to handle missing data in supervised classification problems using 12 datasets in their experiments.

Muñoz and Rueda [22] proposed two imputation methods based on quantiles. The first algorithm is implemented without the aid of auxiliary information and the other is implemented with the aid of auxiliary information. Determining the relationship between the auxiliary variable and the variable of interest is an issue.

Li *et al.* [23] exploited the idea from fuzzy K-means to applying in missing data imputation. The main objective of clustering is dividing the dataset into classes based on objects similarity. The belonging degree of an object to a cluster is determined by the fuzzy membership function. The authors used RMSE error analysis to evaluate the algorithm performance. Depending on the value of the fuzzifier, K-means may outperform fuzzy K-means and vice versa. This indicates that determining the proper value of the fuzzifier parameter is an issue because it is important for the performance of the system.

Batista and Monard [24] use Euclidean distance measure to find the k cases, which have the most similarities. Then, it imputes the missing categorical values in a variable using the most frequent value within the KNN cases. It utilises the unconditional mean for the KNN cases for imputing numerical values. Although KNNI is a simple technique, and its performance is higher than the performance of mean/mode, it is expensive in large dataset because it needs to inspect all cases as many times as the number of cases which contains missingness to find the nearest neighbours of each case with missing value(s).

Honghai *et al.* [25] used SVM to impute missing values. The authors did not compare with any other imputation algorithms. Furthermore, the size of the examples with no missing values,

which will be used in training should be enough; otherwise, the accuracy of regression will be influenced.

Pelckmans *et al.* [26] used a maximum-likelihood technique to get the estimates for the models assumed from their approach for the covariates of missing data. Although the advantage of this approach is that the rules of the classification can be learnt from the data even when the input variables contain missing values, the disadvantage is that the aim of their proposed approach is for high accuracy of classification rather than high accuracy of imputation.

3 Cumulative linear regression

To provide a more in-depth description of the proposed method, this section elaborates the proposed algorithm. List of the terminology used in this work is defined in Table 1.

For any dataset, two cases may occur; the first case when all variables have missing values including the dependent variable and the second case when there is at least one complete variable. Assume that the dependent variable y has no missing values, $X = \{X_i : i = 1, \ldots, n\}$ is set of all predictors, $X^{(\text{Miss})}$ is a set of variables that have missing values, $\{X^{(\text{Miss})} : X^{(\text{Miss})} \subset X\}$, $X^{(\text{Comp})}$ is a set of variables that have no missing values, and $\{X^{(\text{Comp})} : X^{(\text{Comp})} \subset X\}$. The candidate variable from $X^{(\text{Miss})}$ which will be chosen to be the dependent in the first regression iteration will be selected under specific conditions: the candidate variable is highly correlated with the variable y, and the number of all observations, which contain missing values to exist in both the candidate variable and y, is predetermined. This candidate variable will be the dependent variable and y and $X^{(\text{Comp})}$ will be independent in the regression equation. The model is fitted to impute missing values in that variable, then the imputed variable X_{Imp} will be inserted as an independent variable, the independent variables became: y, $X^{(\text{Comp})}$, and $X_{\text{Imp}}^{(\text{Miss})}$. Another variable from $X^{(\text{Miss})}$ will be chosen to be dependent variable, and the model will be fitted again to impute missing values in the variable of interest. This procedure will be repeated until all missing values are imputed. Following is the algorithm, Fig. 1 shows the flowchart of the proposed method (see Fig. 2).

4 Experimental implementation

4.1 Datasets

Five datasets that are commonly used in the literature and different databases repository are used in the comparative study (Table 2). The datasets on hand vary on types and numbers of missing values. This variation is needed for assessing performance and generalisation of the methods. Each dataset is regenerated under the three types of mechanisms, MAR, MCAR, and MNAR, each type with 10, 20, 30, 40, and 50% missingness ratios (MRs).

Table 1 List of terminologies

Terms	Description	Comments
n	number of all variables	
$X^{(\text{Comp})}$	set of complete variables	Comp + Miss = n
$X^{(\text{Miss})}$	set of incomplete variables	
$X_{\text{Imp}}^{(\text{Miss})}$	imputed variable from $X^{(\text{Miss})}$	
c	number of complete independents	$c + m = n$
m	number of variables containing missing variables	
MissObs.y	set of missing observations in the dependent variable y	
MissObs.$X_i^{(\text{Miss})}$	set of missing observations in the independent variable X_i	
α	$\alpha \in [0, 1]$ is an implementation choice	
λ	$\lambda \in [0, 1]$ is an implementation choice	
γ	$\gamma \in [0, 1]$ is an implementation choice	
AllObs	number of all observations	
corr($X_i^{(\text{Miss})}$, y)	correlation between $X_i^{(\text{Miss})}$ and y	
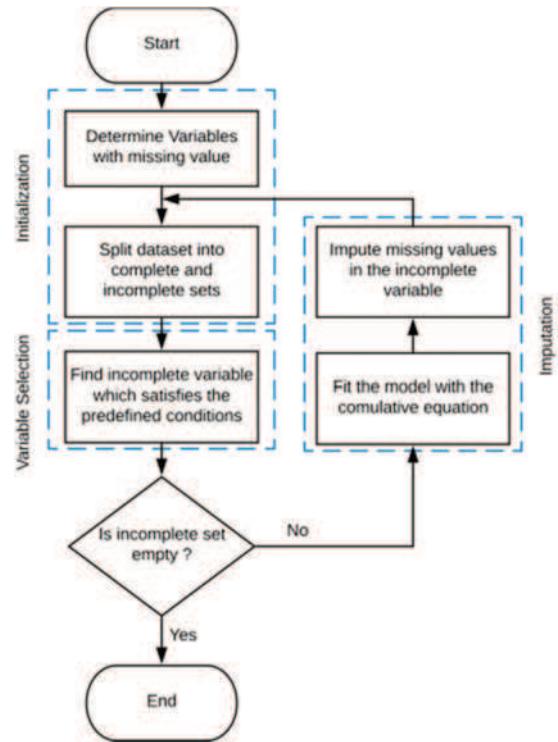
Fig. 1 *Algorithm flowchart*

4.2 R and Python packages

Both R and Python provide some packages to handle missing data. These packages can be structured into missing data exploration (e.g. evaluation with simulations), single imputation, and multiple imputations. The performances of these packages may vary for different datasets depending on the size of the dataset, the mechanism causing the missingness, and the missingness rate. Table 3 clarifies the packages and functions used in the experiments.

4.3 Performance evaluation

The imputation performance of the method is evaluated using RMSE, MAE, R^2, and the time of imputation in seconds (t).

• *RMSE:* Given by (4), in which y_i and $\hat{y}_i$ are the real value and predicted value of the ith observation, respectively, and n is the number of samples [5]

$$\text{RMSE} = \sqrt{\frac{\sum_{i=1}^{n} (y_i - \hat{y}_i)^2}{n}} \qquad (4)$$

• *MAE:* Given by the equation below:

$$\text{MAE}(y, \hat{y}) = \frac{1}{n} \sum_{i=1}^{n} |y_i - \hat{y}_i| \qquad (5)$$

• *R^2:* Given by the equation below:

$$R^2(y, \hat{y}) = 1 - \frac{\sum_{i=1}^{n} (y_i - \hat{y}_i)^2}{\sum_{i=1}^{n} (y_i - \bar{y})^2}; \qquad (6)$$

$$\bar{y} = \frac{1}{n} \sum_{i=1}^{n} y_i$$

• Time of imputation in seconds (t).

- **Initialization**
  - ○ Determine variables with missing values.
  - ○ Split dataset into two subsets
    - ▪ $X^{(Comp)} = \{X_1^{(Comp)}, \ldots, X_c^{(Comp)}\}$, and
    - ▪ $X^{(Miss)} = \{X_1^{(Miss)}, \ldots, X_m^{(Miss)}\}$.
  - ○ $y \in \begin{cases} X^{(Comp)} & if\ MissObs.y = \emptyset \\ X^{(Miss)} & otherwise \end{cases}$
- **Variables selection**
  - ○ From $X^{(Miss)}$, find $X_i^{(Miss)}$, $i \in \{1, \ldots, m\}$, under the following conditions:
    - ▪ $card(MissObs.X_i^{(Miss)} \cup MissObs.y) \leq (\alpha \times AllObs)^1$
    - ▪ $corr(X_i^{(Miss)}, y) \geq \gamma$
  - ○ One of two cases may occur:
    - ▪ Best case:
      - ▪ $(MissObs.X_i^{(Miss)} \cap MissObs.y) = MisObs.X_i^{(Miss)}$

      Or
      - ▪ $(MissObs.X_i^{(Miss)} \cap MissObs.y) = MissObs.y$
    - ▪ Worst case:
      - ▪ $(MissObs.X_i^{(Miss)} \cap MissObs.y) \neq MissObs.X_i^{(Miss)}$

      And
      - ▪ $(MissObs.X_i^{(Miss)} \cap MissObs.y) \neq MissObs.y$
      - ▪ If $card(MissObs.X_i^{(Miss)} \cup MissObs.y) > (\lambda \times AllObs)$

        Select another $X_l^{(Miss)}$, $l \neq i$
- **Imputation**
  - ○ For each column in $X^{(Miss)}$:
    - ▪ Fit the model with cumulative linear regression equation:

    $$X_g^{(Miss)} = \beta_0 + \sum_{i=1}^{c} \beta_i X_i^{(Comp)} + \beta_{c+1} y + \sum_{Imp=1}^{g-1} \beta_{Imp+c+1} X_{Imp}^{(Miss)\ 2}$$

    $$g = 1, 2, \ldots, m$$

    - ▪ Impute missing values.
  - ○ Repeat until all missing values in all columns are imputed.

Fig. 2 *Algorithm: CLR*

Table 2 Datasets specifications

Dataset name	Instances	Features	References
diabetes	424	11	[27]
graduate admissions	500	8	[28]
profit estimation of companies	1000	6	[29]
red & white wine dataset	4898	12	[30]
California	20,640	9	[31]
diamonds	53,940	10	[32]

The imputation method is considered to be efficient if it imputes in a little time with small error and high accuracy. The experiments were carried out using a computer with the following specification: 16 GB memory, Intel Core i5-2400 (3.10 GHz) processor, 1 TB HDD, Gnu/Linux Fedora 28 OS, and Python (version 3.7) and R (version 3.5.2) programming language.

4.4 Experimental results and discussion

4.4.1 Error analysis: Tables 4 and 5 show RMSE and MAE comparisons between cumulative linear regression (CLR) and R packages and CLR and Python packages, respectively. The prominent observations are:

- *For admission dataset:* In MAR, RMSE of the proposed algorithm, CLR, is significantly better than all R packages, KNN, and SoftImpute, and worst than IterativeImputer. MAE of CLR is better than all R packages, except with missForest, better than KNN and SoftImpute, and worst than IterativeImputer.

  In MCAR, RMSE and MAE of CLR are significantly better than all R and Python packages, except with IterativeImputer.

  In MNAR, RMSE of CLR is significantly better than all R and python packages, except with IterativeImputer. MAE of CLR is better than all R and python packages, except missForest, simputation, and IterativeImputer.
- *For diabetes dataset:* RMSE and MAE of CLR are significantly worst than all R packages in all missingness mechanisms, better than KNN and SoftImpute, and worst than IterativeImputer.
- *For-profit dataset:* In MAR, RMSE and MAE of CLR are significantly better than all R packages, except missForest.

Table 3 Packages and functions used for experiments

Function name	Package	Programming language	Description	References
ampute	mice	R	generates missing data in a MNAR, MCAR, or MAR	[33]
mice	mice	R	implements the multivariate imputation by chained equations algorithm	[9, 33]
ForImp	ForImp	R	implements the forward imputation algorithm	[34]
missForest	missForest	R	uses a random forest trained on the observed values to predict the missing values	[35]
Impute_lm	simputation	R	implements regression imputation algorithm	[36]
regressionImp	VIM	R	implements regression imputation algorithm	[37]
IterativeImputer	fancyimpute	Python	implements the multivariate imputation by chained equations algorithm	[38]
KNN	fancyimpute	Python	imputes missing values relying on weights of samples for rows which have observed data	[38]
SoftImpute	fancyimpute	Python	implements spectral regularisation algorithms for learning large incomplete matrices	[38, 39]

In MCAR, RMSE of CLR is better than all R packages, except mice and missForest. MAE of CLR is better than all R packages, except mice, missForest, and simputation.

In MNAR, RMSE of CLR is better than all R packages. MAE of CLR is better than all R packages, except mice and missForest.

RMSE and MAE of CLR are better than SoftImpute and worst than IterativeImputer and KNN in all missingness mechanisms.

• *For wine dataset:* In MAR, RMSE of CLR is significantly better than all R packages. MAE of CLR is better than all R packages, except ForImp and missForest.

In MCAR, RMSE of CLR is significantly better than all R packages. MAE is better than all R packages, except missForest.

In MNAR, RMSE of CLR is significantly better than all R packages. MAE is better than all R packages, except ForImp and missForest.

RMSE and MAE of CLR are better than SoftImpute and worst than IterativeImputer and KNN in all missingness mechanisms.

• *For California dataset:* ForImp failed in imputing the missing values in this dataset. RMSE of CLR is significantly better than all R packages in all missingness mechanisms. MAE of CLR is better than all R packages, except missForest in all missingness mechanisms.

RMSE and MAE of CLR are better than SoftImpute and worst than IterativeImputer in all missingness mechanisms. RMSE and MAE of CLR are better than KNN in MAR and MCAR, and worst in MNAR.

• *For diamond dataset:* ForImp and KNN failed in imputing the missing values in this dataset. RMSE and MAE of CLR are significantly better than all R packages in all missingness mechanisms.

RMSE and MAE of CLR are better than SoftImpute and worst than IterativeImputer in all missingness mechanisms.

4.4.2 Imputation time analysis: Tables 4 and 5 show the imputation times comparisons between CLR and R packages and CLR and Python packages, respectively. The prominent observations are:

• *For admission dataset:* Imputation times of CLR are better than ForImp, mice, and missForest, all Python packages, and worst than simputation and VIM in all missingness mechanisms.
• *For diabetes dataset:* Imputation times of CLR are better than ForImp, IterativeImputer, and SoftImpute, and worst than mice, missForest, simputation, VIM, and KNN in all missingness mechanisms.
• *For-profit dataset:* Imputation times of CLR are better than ForImp, mice, missForest, and all Python packages, and worst than simputation and VIM in all missingness mechanisms.
• *For wine dataset:* Imputation times of CLR are better than ForImp, mice, missForest, KNN, and SoftImpute, worst than simputation and VIM, and behave somewhat similar to IterativeImputer in all missingness mechanisms.

• *For California dataset:* ForImp failed in imputing the missing values in this dataset. Imputation times of CLR are better than mice, and missForest, KNN, and SoftImpute, and worst than simputation, VIM, and IterativeImputer in all missingness mechanisms.
• *For diamond dataset:* ForImp and KNN failed in imputing the missing values in this dataset. Imputation times of CLR are significantly better than mice, missForest, VIM, IterativeImputer, and SoftImpute. CLR behaves somewhat similar to simputation.

4.4.3 Accuracy analysis: The accuracy can be defined as: how well the model will predict the unseen observations. Tables 6 and 7 show the R^2 comparisons between CLR and R packages and CLR and Python packages, respectively. The prominent observations are:

• *For admission dataset:* R^2 of CLR is worst than all R packages, except VIM in all missingness mechanisms. In all missingness mechanisms, R^2 of CLR is worst than IterativeImputer and better than KNN and SoftImpute.
• *For diabetes dataset:* In MAR, R^2 of CLR is better than ForImp and worst than mice, missForest, simputation, and VIM. In MCAR, R^2 of CLR is better than all R packages.

In MNAR, R^2 of CLR is worst than all R packages. R^2 of CLR is worst than IterativeImputer and better than KNN and SoftImpute in all missingness mechanisms.
• *For-profit dataset:* R^2 of CLR is better than VIM and worst than ForImp, mice, missForest, and simputation in all missingness mechanisms.

In MAR, R^2 of CLR is worst than IterativeImputer and better than KNN and SoftImpute. In MCAR, R^2 of CLR is worst than IterativeImputer and better than KNN and SoftImpute. In MNAR, R^2 of CLR is better than IterativeImputer, KNN, and SoftImpute.
• *For wine dataset:* R^2 of CLR is worst than all R packages. R^2 of CLR is worst than IterativeImputer and KNN, and better than SoftImpute in all missingness mechanisms.
• *For California dataset:* ForImp failed in imputing the missing values in this dataset. In MAR and MCAR, R^2 of CLR is better than mice, and VIM, and worst than missForest and simputation.

In MNAR, R^2 of CLR is better than VIM and worst than mice, missForest, and simputation. R^2 of CLR is worst than IterativeImputer and KNN, and better than SoftImpute in all missingness mechanisms.
• *For diamond dataset:* ForImp and KNN failed in imputing the missing values in this dataset. R^2 of CLR is better than VIM, IterativeImputer, and KNN, and worst than mice, missForest, and simputation packages in all missingness mechanisms.

Tables 8 and 9 summarise these observations by comparing the improvements of CLR versus R packages and CLR versus Python packages, respectively.

Table 4 RMSE, MAE, and imputation time (CLR versus R packages)

Admission dataset

Mechanism	% MR	CLR RMSE	CLR MAE	CLR Impute time	ForImp RMSE	ForImp MAE	ForImp Impute time	Mice RMSE	Mice MAE	Mice Impute time	missForest RMSE	missForest MAE	missForest Impute time	Simputation RMSE	Simputation MAE	Simputation Impute time	VIM RMSE	VIM MAE	VIM Impute time
MAR	10	1.05	0.84	0.10	3.08	1.15	2.61	1.99	0.96	0.22	2.40	0.89	0.78	3.01	1.10	0.01	135.63	59.46	0.01
	20	1.11	0.92	0.14	1.69	0.85	3.57	2.37	0.94	0.21	1.46	0.63	0.77	2.52	0.95	0.01	137.15	61.04	0.01
	30	1.03	0.81	0.13	3.69	1.55	4.38	2.56	1.00	0.22	2.09	0.85	0.76	2.06	0.89	0.01	117.49	45.35	0.01
	40	1.10	0.90	0.12	2.47	1.05	4.57	2.66	1.09	0.24	1.73	0.72	0.71	2.30	0.98	0.01	144.41	67.74	0.01
	50	1.07	0.85	0.10	3.36	1.43	4.68	2.55	1.04	0.22	1.85	0.86	0.71	2.02	0.89	0.01	118.63	46.23	0.02
	average	1.07	0.86	0.12	2.86	1.21	3.96	2.42	1.01	0.22	1.91	0.79	0.74	2.38	0.96	0.01	130.66	55.97	0.01
	improvement				167.09%	39.66%	3248.14%	126.44%	16.64%	86.99%	78.10%	−8.63%	528.72%	122.68%	11.59%	−89.86%	12,110.18%	6385.55%	−88.51%
MCAR	10	1.33	1.06	0.11	5.08	2.06	4.57	5.28	2.37	0.21	2.20	1.02	0.78	3.30	1.47	0.01	129.13	55.99	0.01
	20	1.11	0.89	0.13	5.31	2.16	7.06	2.92	1.38	0.23	2.55	1.02	0.74	2.87	1.20	0.01	124.74	51.98	0.01
	30	1.22	0.95	0.10	4.40	1.88	8.26	3.08	1.34	0.22	2.55	1.08	0.73	2.69	1.27	0.01	111.67	41.85	0.01
	40	1.03	0.79	0.07	3.12	1.34	8.69	2.84	1.18	0.22	1.77	0.76	0.69	2.19	0.97	0.01	118.92	46.83	0.01
	50	1.26	1.00	0.11	3.23	1.47	8.44	3.23	1.33	0.22	2.16	0.97	0.71	2.65	1.17	0.01	108.13	39.77	0.01
	average	1.19	0.94	0.10	4.23	1.78	7.40	3.47	1.52	0.22	2.25	0.97	0.73	2.74	1.21	0.01	118.52	47.28	0.01
	improvement				256.03%	89.81%	7005.18%	192.03%	61.83%	111.32%	89.03%	3.24%	599.04%	130.67%	29.24%	−88.68%	9877.11%	4934.63%	−86.56%
MNAR	10	1.00	0.89	0.11	2.27	0.85	4.00	2.22	0.73	0.24	1.86	0.70	0.78	1.58	0.68	0.01	90.79	28.10	0.01
	20	1.30	1.07	0.08	3.17	1.36	6.73	2.70	1.00	0.21	2.51	1.07	0.72	2.35	1.03	0.01	136.23	59.39	0.01
	30	1.04	0.83	0.09	2.37	1.00	8.16	2.83	1.09	0.22	1.77	0.72	0.72	1.66	0.75	0.01	109.93	40.24	0.01
	40	0.95	0.81	0.08	2.35	1.04	8.85	2.56	0.93	0.23	1.64	0.70	0.70	1.58	0.75	0.01	125.11	51.10	0.01
	50	1.18	0.92	0.10	5.44	2.15	8.20	3.37	1.51	0.23	2.30	1.03	0.70	2.60	1.19	0.01	121.31	49.06	0.01
	average	1.09	0.90	0.09	3.12	1.28	7.19	2.74	1.05	0.22	2.02	0.84	0.72	1.96	0.88	0.01	116.68	45.58	0.01
	improvement				185.03%	41.73%	7996.40%	149.89%	16.47%	152.93%	84.07%	−6.81%	711.94%	78.66%	−2.81%	−86.71%	10,557.89%	4943.85%	−84.91%

Diabetes dataset

Mechanism	% MR	CLR RMSE	CLR MAE	CLR Impute time	ForImp RMSE	ForImp MAE	ForImp Impute time	Mice RMSE	Mice MAE	Mice Impute time	missForest RMSE	missForest MAE	missForest Impute time	Simputation RMSE	Simputation MAE	Simputation Impute time	VIM RMSE	VIM MAE	VIM Impute time
MAR	10	282.93	217.35	4.50	0.06	0.05	8.71	0.05	0.03	0.34	0.04	0.03	1.28	0.05	0.04	0.02	0.05	0.04	0.02
	20	258.36	197.66	4.51	0.08	0.06	10.75	0.04	0.03	0.34	0.03	0.02	1.28	0.04	0.04	0.02	0.04	0.04	0.02
	30	251.70	186.61	4.48	0.06	0.04	12.14	0.05	0.03	0.34	0.04	0.03	1.25	0.05	0.04	0.02	0.05	0.04	0.02
	40	247.52	184.20	4.71	0.05	0.04	12.78	0.04	0.02	0.34	0.04	0.03	1.25	0.05	0.04	0.02	0.05	0.04	0.02
	50	255.92	182.74	4.54	0.06	0.04	12.30	0.05	0.03	0.34	0.03	0.03	1.22	0.04	0.04	0.02	0.04	0.04	0.02
	average	259.29	193.71	4.55	0.06	0.05	11.34	0.05	0.03	0.34	0.03	0.03	1.26	0.05	0.04	0.02	0.05	0.04	0.02
	improvement				−99.98%	−99.98%	149.23%	−99.98%	−99.99%	−92.48%	−99.99%	−99.99%	−72.37%	−99.98%	−99.98%	−99.63%	−99.98%	−99.98%	−99.52%
MCAR	10	205.26	152.44	4.55	0.04	0.03	7.78	0.03	0.02	0.34	0.02	0.02	1.27	0.04	0.03	0.02	0.04	0.03	0.02
	20	221.68	153.71	4.58	0.05	0.03	10.35	0.04	0.03	0.34	0.03	0.02	1.26	0.04	0.03	0.02	0.04	0.03	0.02
	30	203.80	153.75	4.53	0.05	0.03	12.20	0.04	0.02	0.34	0.03	0.02	1.26	0.04	0.04	0.02	0.04	0.04	0.02
	40	211.44	154.29	4.64	0.05	0.04	13.01	0.04	0.02	0.34	0.03	0.02	1.27	0.04	0.04	0.02	0.04	0.04	0.02
	50	212.84	153.38	4.60	0.06	0.04	13.15	0.04	0.03	0.35	0.03	0.02	1.22	0.04	0.04	0.02	0.04	0.04	0.02
	average	211.00	153.51	4.58	0.05	0.04	11.30	0.04	0.02	0.34	0.03	0.02	1.26	0.04	0.03	0.02	0.04	0.03	0.02
	improvement				−99.98%	−99.98%	146.75%	−99.98%	−99.98%	−92.59%	−99.99%	−99.99%	−72.57%	−99.98%	−99.98%	−99.64%	−99.98%	−99.98%	−99.52%
MNAR	10	373.20	285.18	4.47	0.07	0.06	7.84	0.05	0.04	0.34	0.04	0.03	1.28	0.06	0.05	0.02	0.06	0.05	0.02
	20	345.05	249.24	4.40	0.05	0.04	10.76	0.04	0.03	0.34	0.04	0.03	1.28	0.06	0.05	0.02	0.06	0.05	0.02
	30	288.17	210.81	4.50	0.07	0.05	12.05	0.04	0.03	0.34	0.03	0.02	1.27	0.05	0.04	0.02	0.05	0.04	0.02
	40	283.05	196.53	4.70	0.07	0.05	12.72	0.04	0.03	0.34	0.03	0.02	1.25	0.05	0.04	0.02	0.05	0.04	0.02
	50	255.00	183.71	4.72	0.08	0.06	13.03	0.05	0.03	0.34	0.04	0.03	1.26	0.05	0.04	0.02	0.05	0.04	0.02
	average	308.89	225.10	4.56	0.07	0.05	11.28	0.04	0.03	0.34	0.04	0.03	1.27	0.05	0.04	0.02	0.05	0.04	0.02
	improvement				−99.98%	−99.98%	147.53%	−99.99%	−99.99%	−92.56%	−99.99%	−99.99%	−72.14%	−99.98%	−99.98%	−99.64%	−99.98%	−99.98%	−99.52%

Profit dataset

Mechanism	% MR	CLR RMSE	CLR MAE	CLR Impute time	ForImp RMSE	ForImp MAE	ForImp Impute time	Mice RMSE	Mice MAE	Mice Impute time	missForest RMSE	missForest MAE	missForest Impute time	Simputation RMSE	Simputation MAE	Simputation Impute time	VIM RMSE	VIM MAE	VIM Impute time
MAR	10	17,236.30	5256.59	0.11	30,759.78	9795.22	37.17	24,588.22	5601.50	0.16	20,103.03	3811.07	1.53	29,326.72	6372.95	0.01	90,010.69	63,068.09	0.01
	20	10,449.68	2688.54	0.08	18,517.88	7056.05	56.12	14,006.26	2731.13	0.17	3591.44	610.74	1.50	17,343.29	2799.70	0.01	82,416.81	53,709.47	0.01
	30	4169.15	1195.68	0.10	15,263.20	7924.13	64.69	8370.57	1689.17	0.19	3562.42	490.37	1.52	8420.67	1478.25	0.01	82,177.12	54,242.73	0.01
	40	4318.42	1158.62	0.10	9867.64	4438.35	62.03	6321.88	1436.00	0.19	3225.20	517.57	1.45	4141.93	1122.21	0.01	83,770.29	56,528.15	0.01
	50	5141.86	1431.08	0.06	13,940.26	6862.12	59.36	5525.70	1380.74	0.20	3284.58	487.53	1.42	6687.70	1285.46	0.01	84,966.43	58,447.89	0.01
	average	8263.08	2346.10	0.09	17,669.75	7215.17	55.87	11,762.53	2567.71	0.18	6753.33	1183.45	1.48	13,184.06	2611.71	0.01	84,668.27	57,199.27	0.01
	improvement				113.84%	207.54%	62,121.38%	42.35%	9.45%	100.00%	−18.27%	−49.56%	1552.56%	59.55%	11.32%	−90.20%	924.66%	2338.06%	−87.31%
MCAR	10	4523.19	1654.02	0.12	12,293.43	6216.98	40.68	6829.08	1897.30	0.17	3629.08	639.25	1.48	8246.99	2298.03	0.01	77,965.22	51,097.82	0.01
	20	20,008.31	4168.56	0.14	16,724.03	6892.49	54.16	13,744.64	3052.00	0.17	6394.16	1102.45	1.47	15,147.34	2716.74	0.01	83,308.11	57,138.95	0.01
	30	2625.59	1045.09	0.08	13,302.60	6091.83	61.72	4094.31	1062.79	0.18	3194.25	523.35	1.53	4246.62	1297.63	0.01	78,826.10	51,461.96	0.01
	40	10,156.11	2263.11	0.17	16,326.38	6901.19	63.49	7281.54	1630.90	0.19	13,451.17	1258.14	1.50	19,363.44	2646.47	0.01	79,275.43	50,702.36	0.01
	50	4625.38	1340.48	0.12	14,510.43	6800.38	57.96	7406.99	1694.99	0.20	4501.30	653.70	1.42	5436.21	1431.34	0.01	77,628.09	50,281.45	0.01
	average	8387.71	2094.25	0.12	14,631.37	6580.57	55.60	7871.31	1867.60	0.18	6233.99	835.38	1.48	10,488.12	2078.04	0.01	79,400.59	52,136.51	0.01
	improvement				74.44%	214.22%	44,814.22%	−6.16%	−10.82%	46.04%	−25.68%	−60.11%	1093.86%	25.04%	−0.77%	−93.21%	846.63%	2389.50%	−91.28%
MNAR	10	25,030.25	7401.38	0.09	15,007.71	6203.73	39.55	16,037.49	2753.41	0.16	26,631.85	4120.40	1.48	39,854.08	6636.58	0.01	100,289.95	75,356.51	0.01
	20	15,999.07	4576.67	0.09	12,607.64	5886.30	53.59	12,768.71	2294.01	0.17	20,045.80	2428.66	1.53	28,269.28	3905.88	0.01	85,662.82	57,484.45	0.01
	30	4569.79	1409.94	0.11	21,293.71	8056.51	63.07	22,588.24	3422.71	0.18	16,215.88	2124.50	1.54	16,210.01	2777.04	0.01	73,190.33	44,075.23	0.01
	40	11,628.64	2101.21	0.10	13,043.30	6190.84	64.48	9958.03	1808.97	0.19	13,244.33	1269.74	1.47	20,398.43	2773.20	0.01	80,045.90	51,451.29	0.01
	50	10,074.35	2336.76	0.07	25,735.41	8198.10	58.56	19,575.13	2706.63	0.22	21,495.37	2009.65	1.37	22,336.02	2865.07	0.01	81,888.06	52,987.73	0.01
	average	13,460.42	3565.19	0.09	17,537.56	6907.10	55.85	16,185.52	2597.15	0.18	19,526.65	2390.59	1.48	25,413.57	3791.55	0.01	84,215.41	56,271.04	0.01
	improvement				30.29%	93.74%	61,003.72%	20.25%	−27.15%	101.09%	45.07%	−32.95%	1516.63%	88.80%	6.35%	−90.81%	525.65%	1478.35%	−87.96%

Wine dataset

Mechanism	% MR	CLR RMSE	CLR MAE	CLR Impute time	ForImp RMSE	ForImp MAE	ForImp Impute time	Mice RMSE	Mice MAE	Mice Impute time	missForest RMSE	missForest MAE	missForest Impute time	Simputation RMSE	Simputation MAE	Simputation Impute time	VIM RMSE	VIM MAE	VIM Impute time
MAR	10	5.83	4.05	0.69	16.45	4.25	3365.29	15.71	5.31	1.54	10.52	2.17	63.47	16.02	5.00	0.03	16.11	5.43	0.05
	20	5.30	4.19	0.80	11.83	3.26	5160.12	12.15	4.25	1.54	7.00	1.98	62.66	14.89	5.19	0.03	14.98	5.62	0.05
	30	4.38	3.42	0.67	11.62	3.23	6208.17	12.36	4.43	2.05	5.61	1.69	61.94	12.83	4.68	0.03	12.94	5.13	0.05
	40	5.00	3.96	0.88	12.84	3.98	6140.06	16.02	5.68	2.04	7.79	2.26	61.81	14.36	5.09	0.03	14.44	5.50	0.05
	50	5.19	3.78	0.70	13.45	4.10	5477.50	13.75	4.92	2.57	8.36	2.18	62.09	13.72	4.67	0.03	13.82	5.13	0.05
	average	5.14	3.88	0.75	13.24	3.76	5270.23	14.00	4.92	1.95	7.86	2.06	62.39	14.37	4.93	0.03	14.46	5.36	0.05
	improvement				157.65%	−3.12%	702,597.04%	172.41%	26.73%	159.68%	52.91%	−47.05%	8219.09%	179.57%	26.95%	−96.32%	181.41%	38.14%	−93.68%
MCAR	10	4.68	3.64	0.83	13.24	3.81	3271.97	14.51	5.39	1.29	6.34	2.09	63.25	13.50	5.36	0.03	13.61	5.86	0.05
	20	4.52	3.64	0.76	12.38	3.47	5359.71	14.34	4.85	1.55	7.06	2.11	62.20	13.08	4.68	0.03	13.19	5.12	0.05
	30	4.79	3.78	0.67	13.00	3.62	6216.06	13.68	4.99	1.82	7.31	2.20	61.67	12.70	4.53	0.03	12.79	4.92	0.05
	40	4.79	3.87	0.73	12.99	3.88	6222.49	13.55	4.90	2.47	6.51	2.11	61.78	13.56	5.03	0.03	13.66	5.48	0.05
	50	4.97	3.76	0.71	14.39	4.38	5495.63	13.27	4.81	2.31	7.38	2.18	60.44	13.44	4.81	0.03	13.56	5.33	0.05
	average	4.75	3.74	0.74	13.20	3.83	5313.17	13.87	4.99	1.89	6.92	2.14	61.87	13.26	4.88	0.03	13.36	5.34	0.05
	improvement				177.95%	2.50%	720,427.75%	192.08%	33.51%	155.93%	45.68%	−42.75%	8290.10%	179.12%	30.55%	−95.90%	181.40%	42.84%	−93.65%
MNAR	10	7.36	5.66	0.64	15.64	4.18	3205.22	15.42	5.70	1.28	10.50	2.89	62.41	18.29	6.77	0.03	18.37	7.21	0.05
	20	5.30	4.33	0.67	11.77	3.70	5452.58	14.58	5.87	1.59	7.53	2.59	62.15	16.50	6.68	0.03	16.60	7.16	0.05
	30	5.83	4.45	0.69	13.97	3.87	6135.49	15.83	5.24	1.80	8.74	2.40	61.50	16.82	6.05	0.03	16.93	6.57	0.05
	40	5.55	4.32	0.68	14.03	3.92	6167.67	15.84	5.47	2.32	9.89	2.65	60.82	16.10	5.49	0.03	16.20	5.96	0.05
	50	5.19	3.94	0.70	14.18	4.34	5425.79	14.27	5.23	2.30	8.07	2.24	60.00	14.85	5.20	0.03	14.96	5.70	0.05
	average	5.85	4.54	0.68	13.92	4.00	5277.35	15.19	5.50	1.86	8.95	2.55	61.38	16.51	6.04	0.03	16.61	6.52	0.05
	improvement				138.03%	−11.89%	781,498.19%	159.80%	21.18%	175.33%	53.03%	−43.74%	8989.96%	182.39%	32.97%	−95.73%	184.15%	43.57%	−93.10%

California dataset

Mechanism	% MR	CLR RMSE	MAE	Impute time	ForImp RMSE	MAE	Impute time	Mice RMSE	MAE	Impute time	missForest RMSE	MAE	Impute time	Simputation RMSE	MAE	Impute time	VIM RMSE	MAE	Impute time
MAR	10	109.90	78.82	1.94	fail	fail	fail	379.93	96.86	7.02	302.17	64.20	832.84	316.85	84.99	0.05	546.21	163.87	0.11
	20	124.88	87.25	1.98	fail	fail	fail	472.62	109.82	11.56	340.21	69.47	831.76	352.01	90.63	0.06	590.74	171.61	0.11
	30	137.24	89.47	1.91	fail	fail	fail	712.75	109.62	15.31	291.67	63.95	800.30	357.23	85.72	0.05	583.52	162.54	0.11
	40	169.56	90.06	1.96	fail	fail	fail	468.93	107.56	19.92	346.31	68.56	783.41	407.09	91.57	0.05	631.09	174.82	0.11
	50	134.36	87.93	1.92	fail	fail	fail	554.96	113.68	25.10	293.40	68.37	743.88	372.34	91.90	0.06	611.90	177.26	0.11
	average	135.19	86.71	1.94				517.84	107.51	15.78	314.75	66.91	798.44	361.10	88.96	0.05	592.69	170.02	0.11
	improvement							283.05%	23.99%	712.80%	132.82%	−22.83%	41,022.71%	167.11%	2.60%	−97.21%	338.42%	96.09%	−94.42%
MCAR	10	158.81	96.93	1.97	fail	fail	fail	526.19	127.58	6.96	333.63	75.14	816.15	431.26	101.61	0.05	665.77	187.06	0.11
	20	231.47	100.64	2.12	fail	fail	fail	727.09	136.19	11.91	621.35	91.15	796.81	679.26	110.42	0.05	881.67	211.09	0.11
	30	148.40	92.69	1.91	fail	fail	fail	498.87	115.51	16.12	318.28	73.64	775.48	384.99	89.99	0.05	636.46	181.91	0.11
	40	177.11	93.16	1.90	fail	fail	fail	604.34	118.69	20.87	423.83	73.97	758.13	492.21	93.08	0.05	694.77	175.76	0.11
	50	130.53	84.90	1.90	fail	fail	fail	508.94	118.63	25.46	308.43	73.10	745.44	366.11	86.54	0.05	621.59	181.79	0.11
	average	169.26	93.66	1.96				573.08	123.32	16.26	401.11	77.40	778.40	470.77	96.33	0.05	700.05	187.52	0.11
	improvement							238.58%	31.66%	730.64%	136.97%	−17.36%	39,655.02%	178.13%	2.84%	−97.31%	313.59%	100.21%	−94.45%
MNAR	10	327.28	213.29	2.02	fail	fail	fail	933.11	227.82	6.42	715.83	160.08	819.82	928.80	219.81	0.05	1282.44	372.71	0.11
	20	238.05	153.30	1.95	fail	fail	fail	676.24	167.59	10.93	510.28	117.70	801.06	661.24	154.25	0.05	997.88	292.38	0.11
	30	216.25	129.36	2.01	fail	fail	fail	640.39	150.20	15.59	464.29	95.33	780.88	575.43	124.27	0.05	881.37	246.62	0.11
	40	215.29	126.72	1.96	fail	fail	fail	678.79	155.35	20.00	459.89	100.09	758.64	587.36	128.93	0.05	889.19	252.69	0.11
	50	205.71	105.94	1.98	fail	fail	fail	659.81	131.63	24.76	489.42	83.73	745.30	564.94	105.04	0.05	828.41	221.06	0.11
	average	240.52	145.72	1.98				717.67	166.52	15.54	527.94	111.38	781.14	663.55	146.46	0.05	975.86	277.09	0.11
	improvement							198.38%	14.27%	683.83%	119.50%	−23.56%	39,303.84%	175.88%	0.50%	−97.33%	305.73%	90.15%	−94.54%

Diamond dataset

Mechanism	% MR	CLR RMSE	MAE	Impute time	ForImp RMSE	MAE	Impute time	Mice RMSE	MAE	Impute time	missForest RMSE	MAE	Impute time	Simputation RMSE	MAE	Impute time	VIM RMSE	MAE	Impute time
MAR	10	0.04	0.03	0.25	fail	fail	fail	810.73	181.39	43.61	287.43	58.11	5296.87	940.07	225.90	0.19	2844.94	764.60	0.29
	20	0.04	0.03	0.22	fail	fail	fail	741.56	158.30	74.10	250.18	50.00	5103.03	814.44	201.63	0.36	2639.72	693.69	0.29
	30	0.04	0.03	0.22	fail	fail	fail	724.57	156.01	106.11	268.93	51.31	4979.90	840.21	199.71	0.17	2573.35	682.66	0.30
	40	0.04	0.03	0.20	fail	fail	fail	694.15	146.29	134.70	252.95	47.17	4874.74	788.86	193.38	0.17	2442.89	634.82	0.30
	50	0.03	0.03	0.21	fail	fail	fail	701.11	140.50	167.77	254.49	44.72	4755.81	818.45	186.28	0.18	2312.34	579.85	0.30
	average	0.04	0.03	0.22				734.43	156.50	105.26	262.80	50.26	5002.07	840.31	201.38	0.21	2562.65	671.13	0.29
	improvement							2,009,173.02%	514,618.45%	48,095.05%	718,864.00%	165,212.78%	2,290,225.92%	2,298,835.97%	662,234.55%	−2.56%	7,010,881.70%	2,207,236.16%	34.62%
MCAR	10	0.03	0.02	0.18	fail	fail	fail	563.28	113.82	42.36	158.82	30.07	5128.06	684.51	165.65	0.17	1833.02	442.77	0.29
	20	0.03	0.03	0.14	fail	fail	fail	597.61	110.84	75.63	235.05	34.88	5156.31	675.49	152.45	0.18	1953.29	453.09	0.29
	30	0.03	0.03	0.16	fail	fail	fail	519.89	99.32	106.79	179.71	31.25	5070.26	636.71	148.80	0.18	1858.04	425.38	0.30
	40	0.03	0.03	0.14	fail	fail	fail	575.88	106.85	136.93	191.95	31.82	4981.96	652.99	153.93	0.18	1900.05	441.53	0.30
	50	0.04	0.03	0.15	fail	fail	fail	555.93	106.68	168.60	190.87	32.78	4876.39	635.46	152.38	0.19	1873.46	441.85	0.30
	average	0.03	0.03	0.15				562.52	107.50	106.06	191.28	32.16	5042.60	657.03	154.64	0.18	1883.57	440.92	0.29
	improvement							1,723,340.53%	405,428.04%	69,954.29%	585,949.09%	121,220.88%	3,330,544.12%	2,012,919.23%	583,250.31%	19.68%	5,770,797.40%	1,663,178.08%	94.32%
MNAR	10	0.04	0.04	0.25	fail	fail	fail	845.72	193.81	41.57	307.43	64.71	5199.54	1167.09	288.98	0.17	3453.65	951.35	0.29
	20	0.04	0.03	0.19	fail	fail	fail	913.42	202.98	74.27	351.32	67.74	5072.84	1075.60	258.04	0.17	3116.00	850.39	0.29
	30	0.04	0.03	0.17	fail	fail	fail	737.42	157.40	107.05	291.28	53.72	5021.21	935.51	209.33	0.18	2735.47	702.12	0.30
	40	0.04	0.03	0.17	fail	fail	fail	738.90	150.97	137.42	277.17	50.06	4894.32	882.23	193.77	0.19	2520.37	639.19	0.30
	50	0.04	0.03	0.18	fail	fail	fail	698.45	144.70	168.01	250.31	46.08	4839.07	815.42	188.76	0.18	2409.69	607.70	0.30
	average	0.04	0.03	0.19				786.78	169.97	105.66	295.50	56.46	5005.40	975.17	227.78	0.18	2847.04	750.15	0.29
	improvement							2,026,021.32%	533,794.03%	55,629.22%	760,880.09%	177,245.41%	2,639,881.43%	2,511,156.92%	715,356.74%	−6.12%	7,331,582.01%	2,356,164.62%	55.38%

Table 5 RMSE, MAE, and imputation time (CLR versus Python packages)

| | | Admission dataset | | | | | | | | | | |
| | | CLR | | | IterativeImputer | | | KNN | | | SoftImpute | | |
Mechanism	% MR	RMSE	MAE	Impute time	RMSE	MAE	Impute time	RMSE	MAE	Impute time	RMSE	MAE	Impute time
MAR	10	1.05	0.84	0.10	1.04	0.83	0.18	1.09	0.82	0.05	50.60	50.53	0.33
	20	1.11	0.92	0.14	1.07	0.88	0.22	1.42	1.14	0.06	48.39	48.29	0.42
	30	1.03	0.81	0.13	1.00	0.79	0.35	1.18	0.80	0.16	49.31	49.22	0.36
	40	1.10	0.90	0.12	1.06	0.88	0.22	1.29	0.98	0.06	49.46	49.35	0.37
	50	1.07	0.85	0.10	1.09	0.88	0.31	1.17	0.92	0.06	48.81	48.72	0.44
	average	1.07	0.86	0.12	1.05	0.85	0.26	1.23	0.93	0.08	49.31	49.22	0.38
	improvement				−1.89%	−1.47%	117.57%	14.74%	7.93%	−32.94%	4508.30%	5604.27%	224.49%
MCAR	10	1.33	1.06	0.11	1.27	1.02	0.19	1.61	1.30	0.07	48.03	47.92	0.33
	20	1.11	0.89	0.13	1.06	0.86	0.21	1.53	1.20	0.06	47.89	47.79	0.35
	30	1.22	0.95	0.10	1.17	0.90	0.24	1.38	1.04	0.07	48.38	48.26	0.39
	40	1.03	0.79	0.07	0.98	0.76	0.17	1.38	1.08	0.07	47.98	47.85	0.38
	50	1.26	1.00	0.11	1.20	0.96	0.21	1.36	1.05	0.06	48.34	48.21	0.33
	average	1.19	0.94	0.10	1.14	0.90	0.20	1.45	1.13	0.06	48.13	48.01	0.36
	improvement				−4.43%	−4.13%	93.28%	22.24%	20.77%	−39.35%	3951.37%	5011.45%	242.23%
MNAR	10	1.00	0.89	0.11	0.98	0.88	0.16	1.15	1.01	0.05	49.92	49.88	0.32
	20	1.30	1.07	0.08	1.26	1.05	0.25	1.15	0.92	0.06	49.40	49.33	0.38
	30	1.04	0.83	0.09	1.02	0.80	0.17	1.08	0.86	0.06	48.76	48.65	0.36
	40	0.95	0.81	0.08	0.93	0.77	0.22	1.20	0.91	0.07	49.48	49.40	0.33
	50	1.18	0.92	0.10	1.16	0.88	0.26	1.46	1.07	0.07	49.26	49.17	0.36
	average	1.09	0.90	0.09	1.07	0.88	0.21	1.21	0.96	0.06	49.36	49.29	0.35
	improvement				−2.21%	−3.09%	138.06%	10.52%	5.69%	−29.28%	4409.12%	5354.16%	295.05%

| | | Diabetes dataset | | | | | | | | | | |
| | | CLR | | | IterativeImputer | | | KNN | | | SoftImpute | | |
Mechanism	% MR	RMSE	MAE	Impute time	RMSE	MAE	Impute time	RMSE	MAE	Impute time	RMSE	MAE	Impute time
MAR	10	0.04	0.03	0.25	0.03	0.02	0.29	0.06	0.05	0.05	0.13	0.11	0.25
	20	0.04	0.03	0.22	0.03	0.02	0.41	0.04	0.04	0.05	0.13	0.11	0.26
	30	0.04	0.03	0.22	0.03	0.02	0.29	0.06	0.05	0.06	0.12	0.10	0.31
	40	0.04	0.03	0.20	0.03	0.02	0.38	0.06	0.05	0.06	0.11	0.09	0.30
	50	0.03	0.03	0.21	0.03	0.02	0.34	0.05	0.04	0.06	0.12	0.10	0.33
	average	0.04	0.03	0.22	0.03	0.02	0.34	0.05	0.04	0.06	0.12	0.10	0.29
	improvement				−26.02%	−28.05%	56.04%	48.89%	44.32%	−74.36%	229.43%	234.39%	34.16%
MCAR	10	0.03	0.02	0.18	0.02	0.02	0.23	0.04	0.04	0.05	0.11	0.10	0.23
	20	0.03	0.03	0.14	0.02	0.02	0.26	0.05	0.04	0.05	0.11	0.09	0.23
	30	0.03	0.03	0.16	0.03	0.02	0.25	0.05	0.04	0.05	0.12	0.11	0.27
	40	0.03	0.03	0.14	0.03	0.02	0.25	0.06	0.04	0.06	0.12	0.10	0.29
	50	0.04	0.03	0.15	0.03	0.02	0.28	0.05	0.04	0.06	0.12	0.10	0.29
	average	0.03	0.03	0.15	0.02	0.02	0.25	0.05	0.04	0.05	0.12	0.10	0.26
	improvement				−24.98%	−23.98%	67.24%	53.68%	54.69%	−65.65%	259.27%	277.80%	73.18%
MNAR	10	0.04	0.04	0.25	0.03	0.03	0.22	0.07	0.06	0.06	0.11	0.10	0.27
	20	0.04	0.03	0.19	0.03	0.02	0.24	0.06	0.05	0.05	0.11	0.09	0.23
	30	0.04	0.03	0.17	0.03	0.02	0.22	0.06	0.05	0.05	0.11	0.09	0.30
	40	0.04	0.03	0.17	0.02	0.02	0.24	0.06	0.04	0.05	0.12	0.10	0.34
	50	0.04	0.03	0.18	0.03	0.02	0.25	0.06	0.04	0.06	0.12	0.09	0.26
	average	0.04	0.03	0.19	0.03	0.02	0.24	0.06	0.05	0.06	0.11	0.10	0.28
	improvement				−29.64%	−30.19%	24.05%	58.23%	58.74%	−70.99%	193.87%	199.37%	47.15%

		Profit dataset											
		CLR			IterativeImputer			KNN			SoftImpute		
Mechanism	% MR	RMSE	MAE	Imputation time	RMSE	MAE	Imputation time	RMSE	MAE	Imputation time	RMSE	MAE	Imputation time
MAR	10	17,236.30	5256.59	0.11	15,725.12	5281.56	0.14	17,916.30	6603.13	0.23	81,701.90	77,047.36	0.11
	20	10,449.68	2688.54	0.08	4645.40	1699.59	0.17	9131.79	1828.25	0.23	77,299.93	73,882.51	0.23
	30	4169.15	1195.68	0.10	4299.23	1305.85	0.16	4190.63	648.12	0.16	73,959.92	70,976.07	0.29
	40	4318.42	1158.62	0.10	4494.01	1075.41	0.14	2767.76	509.52	0.17	73,255.42	70,250.10	0.17
	50	5141.86	1431.08	0.06	4842.25	1432.29	0.15	3029.10	500.82	0.19	74,503.28	71,197.79	0.20
	average	8263.08	2346.10	0.09	6801.21	2158.94	0.15	7407.12	2017.97	0.20	76,144.09	72,670.77	0.20
	improvement				−17.69%	−7.98%	68.37%	−10.36%	−13.99%	120.04%	821.50%	2997.51%	123.83%
MCAR	10	4523.19	1654.02	0.12	6326.34	1894.87	0.14	3756.13	1100.73	0.16	71,122.37	68,458.94	0.13
	20	20,008.31	4168.56	0.14	9214.77	2721.78	0.18	10,305.23	2202.42	0.17	73,804.32	69,807.14	0.24
	30	2625.59	1045.09	0.08	2540.00	871.59	0.14	3213.52	638.01	0.21	66,684.11	62,634.11	0.30
	40	10,156.11	2263.11	0.17	9721.64	1707.58	0.15	12,340.92	1895.57	0.20	73,178.70	68,002.11	0.24
	50	4625.38	1340.48	0.12	4402.36	1231.64	0.22	3102.91	604.21	0.21	70,062.31	67,197.38	0.19
	average	8387.71	2094.25	0.12	6441.02	1685.49	0.17	6543.74	1288.19	0.19	70,970.36	67,219.94	0.22
	improvement				−23.21%	−19.52%	33.44%	−21.98%	−38.49%	52.50%	746.12%	3109.73%	77.54%
MNAR	10	25,030.25	7401.38	0.09	15,972.08	4782.99	0.17	24,555.74	7237.53	0.16	93,776.97	86,254.18	0.27
	20	15,999.07	4576.67	0.09	9500.27	2592.43	0.13	15,944.13	4203.12	0.16	85,581.89	81,046.64	0.16
	30	4569.79	1409.94	0.11	11,104.66	3090.05	0.12	7908.97	2060.19	0.18	80,216.12	76,677.86	0.15
	40	11,628.64	2101.21	0.10	8115.27	1739.53	0.20	11,466.63	1562.81	0.18	79,266.10	75,608.23	0.19
	50	10,074.35	2336.76	0.07	9418.54	1869.45	0.11	10,354.46	1951.50	0.18	78,907.35	74,926.53	0.18
	average	13,460.42	3565.19	0.09	10,822.16	2814.89	0.14	14,045.98	3403.03	0.17	83,549.69	78,902.69	0.19
	improvement				−19.60%	−21.05%	57.33%	4.35%	−4.55%	87.75%	520.71%	2113.14%	107.44%

		Wine dataset											
		CLR			IterativeImputer			KNN			SoftImpute		
Mechanism	% MR	RMSE	MAE	Impute time	RMSE	MAE	Impute time	RMSE	MAE	Impute time	RMSE	MAE	Impute time
MAR	10	5.83	4.05	0.69	5.03	3.15	0.71	5.92	2.88	4.11	23.01	21.73	3.44
	20	5.30	4.19	0.80	4.31	3.32	0.77	4.79	2.78	4.16	21.46	20.34	3.49
	30	4.38	3.42	0.67	3.59	2.83	0.71	4.04	2.46	4.34	20.88	19.98	3.51
	40	5.00	3.96	0.88	4.33	3.28	0.87	4.66	2.93	4.44	21.14	20.15	3.57
	50	5.19	3.78	0.70	4.41	3.12	0.86	5.41	2.93	4.32	21.12	19.91	3.49
	average	5.14	3.88	0.75	4.34	3.14	0.78	4.96	2.79	4.27	21.52	20.42	3.50
	improvement				−15.64%	−19.14%	4.51%	−3.41%	−28.00%	469.89%	318.83%	426.12%	366.51%
MCAR	10	4.68	3.64	0.83	4.28	3.20	0.66	4.39	2.68	4.18	21.22	20.17	3.42
	20	4.52	3.64	0.76	3.73	2.93	0.70	4.16	2.66	4.03	19.40	18.44	3.38
	30	4.79	3.78	0.67	4.02	3.13	0.68	4.64	2.90	4.14	20.16	19.23	3.30
	40	4.79	3.87	0.73	3.70	2.95	0.64	4.46	2.75	4.22	19.53	18.53	3.29
	50	4.97	3.76	0.71	4.15	3.09	0.69	5.37	2.97	4.29	19.99	18.75	3.43
	average	4.75	3.74	0.74	3.97	3.06	0.67	4.60	2.79	4.17	20.06	19.02	3.36
	improvement				−16.30%	−18.09%	−8.76%	−3.08%	−25.35%	465.77%	322.37%	408.85%	356.25%
MNAR	10	7.36	5.66	0.64	6.72	4.86	0.64	6.73	3.72	3.92	25.99	24.88	3.53
	20	5.30	4.33	0.67	4.50	3.63	0.68	4.81	2.88	4.01	23.82	23.04	3.43
	30	5.83	4.45	0.69	4.68	3.42	0.66	4.95	3.02	4.11	23.76	22.76	3.33
	40	5.55	4.32	0.68	4.89	3.61	0.66	5.36	3.13	4.22	23.13	22.16	3.41
	50	5.19	3.94	0.70	4.25	3.16	0.67	4.79	2.75	4.28	22.68	21.75	3.36
	average	5.85	4.54	0.68	5.01	3.74	0.66	5.33	3.10	4.11	23.88	22.92	3.41
	improvement				−14.34%	−17.72%	−2.10%	−8.87%	−31.70%	508.65%	308.38%	404.73%	405.27%

California dataset

Mechanism	% MR	CLR			IterativeImputer			KNN			SoftImpute		
		RMSE	MAE	Impute time	RMSE	MAE	Impute time	RMSE	MAE	Impute time	RMSE	MAE	Impute time
MAR	10	109.90	78.82	1.94	103.80	72.65	1.46	129.11	85.30	68.26	216.32	181.26	9.91
	20	124.88	87.25	1.98	114.42	77.52	1.37	121.40	83.28	69.35	234.01	192.29	9.19
	30	137.24	89.47	1.91	127.52	81.01	1.20	148.96	94.33	71.83	243.57	193.45	8.72
	40	169.56	90.06	1.96	143.79	81.27	1.26	160.79	97.63	72.19	252.19	194.76	9.15
	50	134.36	87.93	1.92	126.48	80.22	1.21	144.62	93.24	77.04	243.47	194.99	9.83
	average	135.19	86.71	1.94	123.20	78.53	1.30	140.98	90.76	71.73	237.91	191.35	9.36
	improvement				−8.87%	−9.43%	−33.02%	4.28%	4.67%	3594.57%	75.99%	120.69%	381.89%
MCAR	10	158.81	96.93	1.97	160.17	87.87	1.85	148.92	92.78	67.64	255.23	199.97	8.98
	20	231.47	100.64	2.12	224.96	94.27	1.22	237.49	101.59	68.65	324.62	215.83	9.23
	30	148.40	92.69	1.91	145.03	85.51	1.19	147.92	93.34	71.08	257.45	206.01	9.03
	40	177.11	93.16	1.90	168.99	87.80	1.17	178.37	96.75	71.53	272.81	199.39	8.98
	50	130.53	84.90	1.90	125.19	83.48	1.19	139.07	91.18	71.99	244.63	202.07	8.79
	average	169.26	93.66	1.96	164.87	87.79	1.32	170.35	95.13	70.18	270.95	204.65	9.00
	improvement				−2.60%	−6.27%	−32.47%	0.64%	1.56%	3484.08%	60.08%	118.50%	359.73%
MNAR	10	327.28	213.29	2.02	304.02	195.79	1.48	291.58	187.73	72.41	479.27	389.57	9.20
	20	238.05	153.30	1.95	217.48	138.78	1.62	220.64	144.10	72.04	383.00	313.79	9.28
	30	216.25	129.36	2.01	198.95	118.69	1.75	207.76	129.37	75.26	353.70	283.35	9.68
	40	215.29	126.72	1.96	201.41	116.81	1.89	201.01	122.30	78.55	344.50	272.21	9.31
	50	205.71	105.94	1.98	196.90	101.12	1.23	202.85	106.13	80.16	325.01	247.55	10.19
	average	240.52	145.72	1.98	223.75	134.24	1.59	224.77	137.93	75.68	377.10	301.30	9.53
	improvement				−6.97%	−7.88%	−19.69%	−6.55%	−5.35%	3717.73%	56.78%	106.76%	380.76%

Diamond dataset

Mechanism	% MR	CLR			IterativeImputer			KNN			SoftImpute		
		RMSE	MAE	Impute time	RMSE	MAE	Impute time	RMSE	MAE	Impute time	RMSE	MAE	Impute time
MAR	10	282.93	217.35	4.50	226.62	144.11	6.07	fail	fail	fail	975.53	797.25	29.51
	20	258.36	197.66	4.51	241.79	183.54	5.17	fail	fail	fail	904.05	721.14	28.88
	30	251.70	186.61	4.48	192.02	123.75	4.18	fail	fail	fail	860.70	676.66	28.75
	40	247.52	184.20	4.71	240.58	176.05	4.28	fail	fail	fail	821.05	635.18	28.27
	50	255.92	182.74	4.54	188.38	116.96	4.10	fail	fail	fail	789.42	598.50	28.97
	average	259.29	193.71	4.55	217.88	148.88	4.76				870.15	685.74	28.87
	improvement				−15.97%	−23.14%	4.68%				235.59%	254.01%	534.89%
MCAR	10	205.26	152.44	4.55	203.71	153.12	4.91	fail	fail	fail	605.01	426.34	28.90
	20	221.68	153.71	4.58	219.28	150.96	5.05	fail	fail	fail	669.41	471.11	28.16
	30	203.80	153.75	4.53	144.41	96.63	6.17	fail	fail	fail	648.41	458.22	30.79
	40	211.44	154.29	4.64	220.69	159.27	6.02	fail	fail	fail	649.21	456.12	29.59
	50	212.84	153.38	4.60	202.56	151.78	4.24	fail	fail	fail	640.02	455.95	29.69
	average	211.00	153.51	4.58	198.13	142.35	5.28				642.41	453.55	29.42
	improvement				−6.10%	−7.27%	15.28%				204.45%	195.44%	542.62%
MNAR	10	373.20	285.18	4.47	384.39	289.11	5.11	fail	fail	fail	1189.35	1001.55	27.67
	20	345.05	249.24	4.40	254.90	170.20	4.20	fail	fail	fail	1051.91	859.63	26.81
	30	288.17	210.81	4.50	277.49	198.49	5.54	fail	fail	fail	951.95	753.66	30.76
	40	283.05	196.53	4.70	212.46	133.60	4.29	fail	fail	fail	879.66	689.55	28.45
	50	255.00	183.71	4.72	245.88	175.59	5.09	fail	fail	fail	821.07	624.78	29.55
	average	308.89	225.10	4.56	275.02	193.40	4.85				978.79	785.83	28.65
	improvement				−10.97%	−14.08%	6.38%				216.87%	249.11%	528.80%

Table 6 R^2 (CLR versus R packages)

	Admission dataset						
Mechanism	% MR	CLR	ForImp	Mice	missForest	Simputation	VIM
MAR	10	0.32	1.00	1.00	1.00	1.00	−0.63
	20	0.56	1.00	1.00	1.00	1.00	−1.09
	30	0.57	1.00	1.00	1.00	1.00	−0.07
	40	0.53	1.00	1.00	1.00	1.00	−1.23
	50	0.51	1.00	1.00	1.00	1.00	−0.28
	average	0.50	1.00	1.00	1.00	1.00	−0.66
	improvement		−0.499681322	−0.499788178	−0.499898657	−0.499792614	1.759968924
MCAR	10	0.42	1.00	1.00	1.00	1.00	−0.09
	20	0.55	1.00	1.00	1.00	1.00	−0.13
	30	0.42	1.00	1.00	1.00	1.00	0.17
	40	0.57	1.00	1.00	1.00	1.00	−0.21
	50	0.51	1.00	1.00	1.00	1.00	0.08
	average	0.49	1.00	1.00	1.00	1.00	−0.04
	improvement		−0.504402848	−0.504611423	−0.504881034	−0.504792024	14.72145471
MNAR	10	0.29	1.00	1.00	1.00	1.00	−0.56
	20	0.34	1.00	1.00	1.00	1.00	−0.55
	30	0.52	1.00	1.00	1.00	1.00	−0.26
	40	0.52	1.00	1.00	1.00	1.00	−0.63
	50	0.42	1.00	1.00	1.00	1.00	0.02
	average	0.42	1.00	1.00	1.00	1.00	−0.39
	improvement		−0.581527339	−0.581622265	−0.581762313	−0.581779554	2.058740903

	Diabetes dataset						
Mechanism	% MR	CLR	ForImp	Mice	missForest	Simputation	VIM
MAR	10	−1.39	−0.37	0.04	0.44	−0.01	−0.01
	20	0.12	−1.88	0.10	0.51	0.11	0.11
	30	0.28	−0.28	0.11	0.42	0.08	0.08
	40	0.23	−0.22	0.38	0.46	0.07	0.07
	50	0.32	−0.42	0.03	0.45	0.14	0.14
	average	−0.09	−0.64	0.13	0.46	0.08	0.08
	improvement		0.863068023	−1.65298308	−1.190310967	−2.131153294	−2.131153294
MCAR	10	0.29	0.00	0.38	0.68	0.07	0.07
	20	0.05	−0.17	−0.08	0.51	0.08	0.08
	30	0.34	0.02	0.32	0.55	0.08	0.08
	40	0.33	−0.24	0.29	0.52	0.06	0.06
	50	0.38	−0.43	0.16	0.49	0.17	0.17
	average	0.28	−0.16	0.22	0.55	0.09	0.09
	improvement		2.709995375	0.289182067	−0.495628514	2.058709206	2.058709206
MNAR	10	−16.36	−1.16	−0.15	0.29	−0.38	−0.38
	20	−0.08	−0.30	0.26	0.40	−0.46	−0.46
	30	0.07	−1.30	0.22	0.43	−0.12	−0.12
	40	0.25	−1.75	0.09	0.52	−0.08	−0.08
	50	0.17	−1.66	0.14	0.41	−0.05	−0.05
	average	−3.19	−1.23	0.11	0.41	−0.22	−0.22
	improvement		−1.585414155	−29.8089156	−8.766413138	−13.54237715	−13.54237715

	Profit dataset						
Mechanism	% MR	CLR	ForImp	Mice	missForest	Simputation	VIM
MAR	10	0.34	0.83	0.89	0.93	0.84	−0.50
	20	0.45	0.93	0.96	1.00	0.94	−0.42
	30	0.49	0.95	0.98	1.00	0.98	−0.56
	40	0.44	0.98	0.99	1.00	1.00	−0.72
	50	0.48	0.96	0.99	1.00	0.99	−0.64
	average	0.44	0.93	0.96	0.98	0.95	−0.57
	improvement		−0.522138714	−0.540281904	−0.549666029	−0.533804894	1.779161601
MCAR	10	0.50	0.96	0.99	1.00	0.98	−0.58
	20	0.32	0.94	0.96	0.99	0.95	−0.61
	30	0.49	0.95	1.00	1.00	1.00	−0.61
	40	−1.36	0.94	0.99	0.96	0.91	−0.44
	50	0.49	0.94	0.99	0.99	0.99	−0.58
	average	0.09	0.95	0.98	0.99	0.97	−0.56
	improvement		−0.905889058	−0.909330413	−0.90977706	−0.907779756	1.157710399
MNAR	10	0.06	0.97	0.96	0.90	0.78	−0.42
	20	0.30	0.97	0.97	0.93	0.86	−0.30
	30	0.23	0.90	0.89	0.94	0.94	−0.13
	40	0.31	0.96	0.98	0.96	0.91	−0.32
	50	0.34	0.87	0.92	0.91	0.90	−0.34
	average	0.25	0.94	0.95	0.93	0.88	−0.30
	improvement		−0.735836311	−0.738816076	−0.734014255	−0.718824116	1.822865615

				Wine dataset			
Mechanism	% MR	CLR	ForImp	Mice	missForest	Simputation	VIM
MAR	10	0.25	0.87	0.88	0.95	0.88	0.88
	20	0.29	0.92	0.92	0.97	0.88	0.88
	30	0.06	0.92	0.91	0.98	0.91	0.91
	40	0.27	0.92	0.87	0.97	0.90	0.90
	50	0.30	0.90	0.89	0.96	0.89	0.89
	average	0.23	0.91	0.90	0.97	0.89	0.89
	improvement		−0.742723998	−0.739732287	−0.758591179	−0.738156952	−0.73773539
MCAR	10	0.30	0.91	0.89	0.98	0.90	0.90
	20	0.32	0.90	0.87	0.97	0.89	0.89
	30	0.29	0.90	0.89	0.97	0.91	0.91
	40	0.31	0.90	0.89	0.97	0.89	0.89
	50	0.30	0.88	0.89	0.97	0.89	0.89
	average	0.30	0.90	0.89	0.97	0.90	0.90
	improvement		−0.660673377	−0.656693278	−0.68635456	−0.660400713	−0.65976529
MNAR	10	−0.17	0.90	0.90	0.96	0.86	0.86
	20	0.02	0.95	0.92	0.98	0.90	0.90
	30	0.07	0.92	0.90	0.97	0.88	0.88
	40	0.15	0.91	0.89	0.96	0.89	0.89
	50	0.21	0.91	0.91	0.97	0.90	0.90
	average	0.06	0.92	0.90	0.97	0.89	0.88
	improvement		−0.938695666	−0.937705759	−0.941740854	−0.936494216	−0.936392402

				California dataset			
Mechanism	% MR	CLR	ForImp	Mice	missForest	Simputation	VIM
MAR	10	0.33	fail	0.48	0.67	0.64	−0.08
	20	0.30	fail	0.31	0.64	0.62	−0.07
	30	0.34	fail	−0.59	0.73	0.60	−0.07
	40	0.31	fail	0.41	0.68	0.56	−0.07
	50	0.35	fail	0.12	0.75	0.60	−0.07
	average	0.33		0.15	0.70	0.60	−0.07
	improvement			1.248453887	−0.530381119	−0.457971364	5.534425731
MCAR	10	0.32	fail	0.33	0.73	0.55	−0.07
	20	0.35	fail	0.28	0.48	0.38	−0.05
	30	0.31	fail	0.34	0.73	0.61	−0.07
	40	0.31	fail	0.20	0.61	0.47	−0.06
	50	0.30	fail	0.28	0.73	0.63	−0.08
	average	0.32		0.29	0.66	0.53	−0.07
	improvement			0.105802803	−0.516172661	−0.396280349	5.848101524
MNAR	10	0.08	fail	0.43	0.66	0.43	−0.08
	20	0.11	fail	0.50	0.72	0.52	−0.08
	30	0.20	fail	0.43	0.70	0.54	−0.07
	40	0.21	fail	0.37	0.71	0.53	−0.08
	50	0.25	fail	0.32	0.63	0.50	−0.07
	average	0.17		0.41	0.68	0.51	−0.08
	improvement			−0.589136901	−0.752866272	−0.666131058	3.196220474

				Diamond dataset			
Mechanism	% MR	CLR	ForImp	Mice	missForest	Simputation	VIM
MAR	10	0.34	fail	0.91	0.99	0.88	−0.08
	20	0.45	fail	0.92	0.99	0.90	−0.08
	30	0.49	fail	0.91	0.99	0.89	−0.08
	40	0.44	fail	0.91	0.99	0.89	−0.07
	50	0.48	fail	0.90	0.99	0.87	−0.07
	average	0.44		0.91	0.99	0.88	−0.08
	improvement			−0.514311739	−0.552252118	−0.49914954	6.875374398
MCAR	10	0.50	fail	0.90	0.99	0.85	−0.06
	20	0.32	fail	0.90	0.98	0.87	−0.06
	30	0.49	fail	0.92	0.99	0.88	−0.06
	40	−1.36	fail	0.90	0.99	0.87	−0.06
	50	0.49	fail	0.91	0.99	0.88	−0.06
	average	0.09		0.91	0.99	0.87	−0.06
	improvement			−0.901590986	−0.909911626	−0.897674375	2.480163726
MNAR	10	0.06	fail	0.94	0.99	0.88	−0.08
	20	0.30	fail	0.91	0.99	0.87	−0.08
	30	0.23	fail	0.92	0.99	0.87	−0.07
	40	0.31	fail	0.91	0.99	0.87	−0.07
	50	0.34	fail	0.91	0.99	0.88	−0.07
	average	0.25		0.92	0.99	0.87	−0.08
	improvement			−0.73035659	−0.749937643	−0.717146717	4.263328393

CAAI Trans. Intell. Technol., 2019, Vol. 4, Iss. 3, pp. 182–200

194

Table 7 R^2 (CLR versus Python packages)

			Admission dataset		
Mechanism	% MR	CLR	IterativeImputer	KNN	SoftImpute
MAR	10	0.32	0.35	0.21	−366.24
	20	0.56	0.62	0.44	−180.61
	30	0.57	0.68	0.54	−208.49
	40	0.53	0.60	0.48	−171.38
	50	0.51	0.54	0.34	−186.17
	average	0.50	0.56	0.40	−222.58
	improvement		−0.106621426	0.2431064	1.002246098
MCAR	10	0.42	0.52	0.08	−266.78
	20	0.55	0.64	0.26	−252.19
	30	0.42	0.49	0.22	−138.09
	40	0.57	0.65	0.57	−158.53
	50	0.51	0.60	0.51	−205.66
	average	0.49	0.58	0.33	−204.25
	improvement		−0.147779807	0.508159912	1.002423192
MNAR	10	0.29	0.24	0.17	−58.43
	20	0.34	0.47	0.31	−274.99
	30	0.52	0.61	0.55	−183.30
	40	0.52	0.59	0.33	−301.32
	50	0.42	0.47	0.36	−231.15
	average	0.42	0.48	0.34	−209.84
	improvement		−0.12093135	0.218513485	1.001992277

			Diabetes dataset		
Mechanism	% MR	CLR	IterativeImputer	KNN	SoftImpute
MAR	10	−1.39	−1.29	−9.70	−30.45
	20	0.12	0.40	−0.43	−20.40
	30	0.28	0.55	−0.77	−8.86
	40	0.23	0.53	−0.55	−9.00
	50	0.32	0.56	−0.47	−10.11
	average	−0.09	0.15	−2.38	−15.76
	improvement		−1.588102117	0.963530381	0.994481864
MCAR	10	0.29	0.29	−0.53	−23.16
	20	0.05	0.55	−0.92	−15.09
	30	0.34	0.54	−0.41	−10.44
	40	0.33	0.61	−0.85	−13.34
	50	0.38	0.57	−0.41	−12.84
	average	0.28	0.56	−0.65	−12.93
	improvement		−0.512670875	1.425309053	1.021283889
MNAR	10	−16.36	−2.47	−136.71	−1641.60
	20	−0.08	0.44	−1.28	−17.00
	30	0.07	0.34	−1.44	−10.75
	40	0.25	0.53	−0.83	−13.42
	50	0.17	0.47	−0.97	−21.66
	average	0.10	0.44	−1.13	−15.71
	improvement		−0.766238073	1.091965758	1.006615723

			Profit dataset		
Mechanism	% MR	CLR	IterativeImputer	KNN	SoftImpute
MAR	10	0.34	0.50	−0.10	−9.57
	20	0.52	0.66	0.23	−141.90
	30	0.56	0.56	0.17	−99.53
	40	0.56	0.59	0.16	−47.09
	50	0.51	0.54	0.26	−66.90
	average	0.50	0.57	0.15	−73.00
	improvement		−0.220871018	2.046321931	1.006063819
MCAR	10	0.50	0.54	0.23	−28.42
	20	0.06	0.49	−0.03	−19.75
	30	0.58	0.62	−0.09	−72.39
	40	0.54	0.30	0.05	−16.66
	50	0.50	0.52	0.18	−27.06
	average	0.44	0.50	0.07	−32.86
	improvement		−0.820804669	0.321239445	1.002711775
MNAR	10	0.00	0.23	−0.48	−8.48
	20	−0.01	0.32	−0.45	−61.43
	30	0.34	0.18	−0.13	−10.82
	40	0.34	0.53	−0.05	−11.40
	50	0.45	−365.87	0.19	−67.93
	average	0.22	−72.92	−0.18	−32.01
	improvement		1.003388692	2.359632511	1.007719285

Wine dataset

Mechanism	% MR	CLR	IterativeImputer	KNN	SoftImpute
MAR	10	0.25	0.45	0.27	−13,775.72
	20	0.29	0.43	0.32	−12,544.84
	30	0.06	0.10	0.33	−14,769.83
	40	0.27	0.48	0.24	−13,008.70
	50	0.30	0.33	0.31	−13,071.33
	average	0.23	0.36	0.30	−13,434.09
	improvement		−0.350381102	−0.211157826	1.000017371
MCAR	10	0.30	0.51	0.36	−9895.94
	20	0.32	0.55	0.46	−11,519.12
	30	0.29	0.40	0.32	−11,734.15
	40	0.31	0.55	0.32	−11,988.76
	50	0.30	0.52	0.26	−12,346.21
	average	0.30	0.51	0.34	−11,496.84
	improvement		−0.398942539	−0.113836597	1.000026515
MNAR	10	−0.17	0.21	0.08	−17,674.77
	20	0.02	0.23	0.13	−10,815.26
	30	0.07	0.29	0.18	−12,627.38
	40	0.15	0.43	0.22	−15,926.66
	50	0.21	0.48	0.30	−13,380.28
	average	0.06	0.33	0.18	−14,084.87
	improvement		−0.829087535	−0.690995105	1.000003995

California dataset

Mechanism	% MR	CLR	IterativeImputer	KNN	SoftImpute
MAR	10	0.33	−0.30	0.28	−461.26
	20	0.30	0.44	0.47	−483.52
	30	0.34	0.56	0.46	−468.80
	40	0.31	0.55	0.50	−481.02
	50	0.35	0.54	0.46	−466.58
	average	0.33	0.36	0.43	−472.24
	improvement		−0.093047205	−0.246243669	1.000691938
MCAR	10	0.32	−0.14	0.39	−474.00
	20	0.35	0.52	0.43	−492.32
	30	0.31	0.56	0.40	−484.43
	40	0.31	0.48	0.46	−472.15
	50	0.30	0.45	0.44	−488.76
	average	0.32	0.37	0.42	−482.33
	improvement		−0.14908232	−0.250245491	1.000658548
MNAR	10	0.08	0.45	0.18	−891.06
	20	0.11	0.48	0.26	−928.18
	30	0.20	0.51	0.33	−789.17
	40	0.21	0.52	0.39	−722.22
	50	0.25	0.39	0.41	−706.41
	average	0.17	0.47	0.31	−807.41
	improvement		−0.639075761	−0.459159797	1.000209402

Diamond dataset

Mechanism	% MR	CLR	IterativeImputer	KNN	SoftImpute
MAR	10	0.343801224	−487.387274	fail	−355.263871
	20	0.45	−16.10	fail	−287.03
	30	0.49	0.40	fail	−297.30
	40	0.44	−1841.19	fail	−273.03
	50	0.48	0.53	fail	−290.06
	average	0.44	−468.75		−300.54
	improvement		1.000944323		1.001472868
MCAR	10	0.50	−618.53	fail	−303.26
	20	0.32	−324.49	fail	−291.29
	30	0.49	0.50	fail	−286.22
	40	−1.36	−950.67	fail	−323.68
	50	0.49	−2687.05	fail	−291.03
	average	0.09	−916.05		−299.10
	improvement		1.000097262		1.000297885
MNAR	10	0.06	−486.03	fail	−177.24
	20	0.30	0.31	fail	−281.27
	30	0.23	−25.54	fail	−289.89
	40	0.31	0.37	fail	−332.48
	50	0.34	−487.39	fail	−355.26
	average	0.25	−199.66		−287.23
	improvement		1.001237689		1.000860328

Table 8 CLR versus R packages

Admission dataset

Mechanism	ForImp				Mice				missForest				Simputation				VIM			
	RMSE	MAE	Impute time	R2	RMSE	MAE	Impute time	R2	RMSE	MAE	Impute time	R2	RMSE	MAE	Impute time	R2	RMSE	MAE	Impute time	R2
MAR	better	better	better	worst	better	better	better	worst	better	worst	better	worst	better	better	worst	worst	better	better	worst	better
MCAR	better	better	better	worst	better	better	better	worst	better	better	better	worst	better	better	worst	worst	better	better	worst	better
MNAR	better	better	better	worst	better	better	better	worst	better	worst	better	worst	better	worst	worst	worst	better	better	worst	better

Diabetes dataset

Mechanism	ForImp				Mice				missForest				Simputation				VIM			
	RMSE	MAE	Impute time	R2	RMSE	MAE	Impute time	R2	RMSE	MAE	Impute time	R2	RMSE	MAE	Impute time	R2	RMSE	MAE	Impute time	R2
MAR	worst	worst	better	better	worst	worst	worst	worst	worst	worst	worst	worst	worst	worst	worst	worst	worst	worst	worst	worst
MCAR	worst	worst	better	better	worst	worst	worst	worst	better	worst	worst	worst	worst	worst	worst	better	worst	worst	worst	better
MNAR	worst	worst	better	worst	worst	worst	worst	worst	worst	worst	worst	worst	worst	worst	worst	worst	worst	worst	worst	worst

Profit dataset

Mechanism	ForImp				Mice				missForest				Simputation				VIM			
	RMSE	MAE	Impute time	R2	RMSE	MAE	Impute time	R2	RMSE	MAE	Impute time	R2	RMSE	MAE	Impute time	R2	RMSE	MAE	Impute time	R2
MAR	better	better	better	worst	better	better	better	worst	worst	worst	better	worst	better	better	worst	worst	better	better	worst	better
MCAR	better	better	better	worst	worst	worst	better	worst	worst	worst	better	worst	better	worst	worst	worst	better	better	worst	better
MNAR	better	better	better	worst	better	worst	better	worst	better	worst	better	worst	better	better	worst	worst	better	better	worst	better

Wine dataset

Mechanism	ForImp				Mice				missForest				Simputation				VIM			
	RMSE	MAE	Impute time	R2	RMSE	MAE	Impute time	R2	RMSE	MAE	Impute time	R2	RMSE	MAE	Impute time	R2	RMSE	MAE	Impute time	R2
MAR	better	worst	better	worst	better	better	better	worst	better	worst	better	worst	better	better	worst	worst	better	better	worst	worst
MCAR	better	better	better	worst	better	better	better	worst	better	worst	better	worst	better	better	worst	worst	better	better	worst	worst
MNAR	better	worst	better	worst	better	better	better	worst	better	worst	better	worst	better	better	worst	worst	better	better	worst	worst

Diamond dataset

Mechanism	ForImp				Mice				missForest				Simputation				VIM			
	RMSE	MAE	Impute time	R2	RMSE	MAE	Impute time	R2	RMSE	MAE	Impute time	R2	RMSE	MAE	Impute time	R2	RMSE	MAE	Impute time	R2
MAR	better	better	better	better	better	better	better	worst	better	better	better	worst	better	better	worst	worst	better	better	better	better
MCAR	better	better	better	better	better	better	better	worst	better	better	better	worst	better	better	worst	worst	better	better	better	better
MNAR	better	better	better	better	better	better	better	worst	better	better	better	worst	better	better	worst	worst	better	better	better	better

California dataset

Mechanism	ForImp				Mice				missForest				Simputation				VIM			
	RMSE	MAE	Impute time	R2	RMSE	MAE	Impute time	R2	RMSE	MAE	Impute time	R2	RMSE	MAE	Impute time	R2	RMSE	MAE	Impute time	R2
MAR	better	better	better	better	better	better	better	better	better	worst	better	worst	better	better	worst	worst	better	better	worst	better
MCAR	better	better	better	better	better	better	better	better	better	worst	better	worst	better	better	worst	worst	better	better	worst	better
MNAR	better	better	better	better	better	better	better	worst	better	worst	better	worst	better	better	worst	worst	better	better	worst	better

Table 9 CLR versus Python packages

Admission dataset

Mechanism	IterativeImputer				KNN				SoftImpute			
	RMSE	MAE	Impute time	R2	RMSE	MAE	Impute time	R2	RMSE	MAE	Impute time	R2
MAR	worst	worst	better	worst	better	better	worst	better	better	better	better	better
MCAR	worst	worst	better	worst	better	better	worst	better	better	better	better	better
MNAR	worst	worst	better	worst	better	better	worst	better	better	better	better	better

Diabetes dataset

Mechanism	IterativeImputer				KNN				SoftImpute			
	RMSE	MAE	Impute time	R2	RMSE	MAE	Impute time	R2	RMSE	MAE	Impute time	R2
MAR	worst	worst	better	worst	better	better	worst	better	better	better	better	better
MCAR	worst	worst	better	worst	better	better	worst	better	better	better	better	better
MNAR	worst	worst	better	worst	better	better	worst	better	better	better	better	better

Profit dataset

Mechanism	IterativeImputer				KNN				SoftImpute			
	RMSE	MAE	Impute time	R2	RMSE	MAE	Impute time	R2	RMSE	MAE	Impute time	R2
MAR	worst	worst	better	worst	worst	worst	better	better	better	better	better	better
MCAR	worst	worst	better	worst	worst	worst	better	better	better	better	better	better
MNAR	worst	worst	better	better	better	worst	better	better	better	better	better	better

Wine dataset

Mechanism	IterativeImputer				KNN				SoftImpute			
	RMSE	MAE	Impute time	R2	RMSE	MAE	Impute time	R2	RMSE	MAE	Impute time	R2
MAR	worst	worst	better	worst	worst	worst	better	worst	better	better	better	better
MCAR	worst	worst	worst	worst	worst	worst	better	worst	better	better	better	better
MNAR	worst	worst	worst	worst	worst	worst	better	worst	better	better	better	better

California dataset

Mechanism	IterativeImputer				KNN				SoftImpute			
	RMSE	MAE	Impute time	R2	RMSE	MAE	Impute time	R2	RMSE	MAE	Impute time	R2
MAR	worst	worst	worst	worst	better	better	better	worst	better	better	better	better
MCAR	worst	worst	worst	worst	better	better	better	worst	better	better	better	better
MNAR	worst	worst	worst	worst	worst	worst	better	worst	better	better	better	better

Diamond dataset

Mechanism	IterativeImputer				KNN				SoftImpute			
	RMSE	MAE	Impute time	R2	RMSE	MAE	Impute time	R2	RMSE	MAE	Impute time	R2
MAR	worst	worst	better	better	better	better	better	better	better	better	better	better
MCAR	worst	worst	better	better	better	better	better	better	better	better	better	better
MNAR	worst	worst	better	better	better	better	better	better	better	better	better	better

5 Conclusion, findings, and future work

The quality of the data has a significant impact on the statistical analysis. Dealing with missing values in the dataset is an important step in the data preprocessing stage. Therefore, it has magnitude weightiness in data analysis. In addition to providing an overview of the studies related to dealing with missing data, an imputation method has been proposed in this paper to improve the quality of the data by exploiting all available variables. Correlation between the variable of interest, which contains missing values, and the candidate variable, which will be used in the imputation, and the number of missing values in both of them are two important factors to be taken into account when choosing this candidate variable. The imputed variable will be a candidate variable for imputing another incomplete variable. The findings of the proposed method make it easy to implement, work with any dataset and does not fail in the imputation regardless of the size of the dataset or the missingness mechanism. However, the proposed method is not the optimum one, at worst; it behaves somewhat similar to the common methods. In future research avenues, the proposed imputation approach will be analysed in other datasets, other units of standard error (e.g. T-value and P-value) will be taken into consideration when selecting the candidate variable. The most important future trend is to take advantage of algorithms that deal with optimisation problems with mixed variables such as GSA-GA algorithm [40].

6 References

[1] Kang, H.: 'The prevention and handling of the missing data', *Korean J. Anesthesiol.*, 2013, **64**, (5), pp. 402–406
[2] Cismondi, F., Fialho, A.S., Vieira, S.M., *et al.*: 'Missing data in medical databases: impute, delete or classify?', *Artif. Intell. Med.*, 2013, **58**, (1), pp. 63–72
[3] Hapfelmeier, A., Hothorn, T., Ulm, K., *et al.*: 'A new variable importance measure for random forests with missing data', *Stat. Comput.*, 2014, **24**, (1), pp. 21–34
[4] Batista, G., Monard, M.C.: 'A study of K-nearest neighbour as an imputation method'. HIS'02 Second Int. Conf. Hybrid Intelligent Systems, Santiago, Chile, December 2002, pp. 251–260
[5] Aydilek, I.B., Arslan, A.: 'A hybrid method for imputation of missing values using optimized fuzzy c-means with support vector regression and a genetic algorithm', *Inf. Sci. (New York)*, 2013, **233**, pp. 25–35
[6] Pampaka, M., Hutcheson, G., Williams, J.: 'Handling missing data: analysis of a challenging dataset using multiple imputation', *Int. J. Res. Method Educ.*, 2016, **39**, (1), pp. 19–37
[7] Abdella, M., Marwala, T.: 'The use of genetic algorithms and neural networks to approximate missing data in database'. IEEE Third Int. Conf. Computational Cybernetics (ICCC) 2005, Mauritius, 2005, pp. 207–212
[8] Luengo, J., García, S., Herrera, F.: 'On the choice of the best imputation methods for missing values considering three groups of classification methods', *Knowl. Inf. Syst.* 2012, **32**, (1), pp. 77–108
[9] Donders, R., van der Heijden, G., Stijnen, T., *et al.*: 'Review: a gentle introduction to imputation of missing values', *J. Clin. Epidemiol.*, 2006, **59**, (10), pp. 1087–1091
[10] Huque, M.H., Carlin, J.B., Simpson, J.A., *et al.*: 'A comparison of multiple imputation methods for missing data in longitudinal studies', *BMC Med. Res. Methodol.*, 2018, **18**, (1), pp. 1–16
[11] Horton, N.J., Kleinman, K.P.: 'A comparison of missing data methods and software to fit incomplete data regression models', *Am. Stat.*, 2007, **61**, (1), pp. 79–90
[12] Kalkan, Ö.K., Kara, Y., Kelecioğlu, H.: 'Evaluating performance of missing data imputation methods in IRT analyses', *Int. J. Assess. Tools Educ.*, 2018, **5**, (3), pp. 403–416
[13] Gelman, A., Hill, J.: 'Data analysis using regression and multilevel/hierarchical models' (Cambridge University Press, Cambridge, 2006)
[14] Farhangfar, A., Kurgan, L.A., Pedrycz, W.: 'A novel framework for imputation of missing values in databases', *IEEE Trans. Syst. Man Cybern. A, Syst. Hum.*, 2007, **37**, (5), pp. 692–709
[15] Little, R., Rubin, D.: 'Statistical analysis with missing data' (John Wiley & Sons, Hoboken, 2014, 2nd edn.)
[16] Royston, P.: 'Multiple imputation of missing values', *Stat. J.*, 2004, **4**, (3), pp. 227–241
[17] Allison, P.D.: 'Handling missing data by maximum likelihood', 2012
[18] Scheffer, J.: 'Dealing with missing data', *Res. Lett. Inf. Math. Sci.*, 2002, **3**, pp. 153–160
[19] Qin, Y., Zhang, S., Zhu, X., *et al.*: 'Semi-parametric optimization for missing data imputation', *Appl. Intell.*, 2007, **27**, (1), pp. 79–88
[20] Chen, C., Twycross, J., Garibaldi, J.M.: 'A new accuracy measure based on bounded relative error for time series forecasting', *PLOS One*, 2017, **12**, (3), pp. 1–23
[21] Acuña, E., Rodriguez, C.: 'The treatment of missing values and its effect on classifier accuracy'. Classification, Clustering, and Data Mining Applications, Berlin, Germany, 2004, pp. 639–648
[22] Muñoz, J.F., Rueda, M.: 'New imputation methods for missing data using quantiles', *J. Comput. Appl. Math.*, 2009, **232**, (2), pp. 305–317
[23] Li, D., Deogun, J., Spaulding, W., *et al.*: 'Towards missing data imputation: a study of fuzzy k-means clustering method'. Int. Conf. Rough Sets and Current Trends in Computing, Berlin, Heidelberg, 2004, pp. 573–579
[24] Batista, G.E.A.P.A., Monard, M.C.: 'An analysis of four missing data treatment methods for supervised learning', *Appl. Artif. Intell.*, 2003, **17**, (5–6), pp. 519–533
[25] Honghai, F., Guoshun, C., Cheng, Y., *et al.*: 'A SVM regression based approach to filling in missing values'. Int. Conf. Knowledge-Based Intelligent Information and Engineering Systems, Berlin, Heidelberg, 2005, pp. 581–587
[26] Pelckmans, K., De Brabanter, J., Suykens, J.A.K., *et al.*: 'Handling missing values in support vector machine classifiers', *Neural Netw.*, 2005, **18**, (5–6), pp. 684–692
[27] Statistics, N. D. of: 'Diabetes Data-1-5-2019'. Available at https://www4.stat.ncsu.edu/~boos/var.select/diabetes.html, accessed 01 May 2019
[28] Acharya, M.S.: 'Graduate admissions-1-5-2019', 2018. Available at https://www.kaggle.com/mohansacharya/graduate-admissions, accessed 01 May 2019
[29] Boosuro: 'profit estimation', 2018. Available at https://github.com/boosuro/profit_estimation_of_companies, accessed 17 February 2019
[30] Patnaik, K.: 'Red & white wine', 2017. Available at https://www.kaggle.com/numberswithkartik/red-white-wine-dataset, accessed 15 February 2019
[31] Nugent, C.: 'California housing prices-3-5-2019', 2017. Available at https://www.kaggle.com/camnugent/california-housing-prices, accessed 03 May 2019
[32] Shiva, M.: 'Diamonds', 2017. Available at https://www.kaggle.com/shivam2503/diamonds, accessed 10 February 2019
[33] van Buuren, S., Groothuis-Oudshoorn, K., Robitzsch, A., *et al.*: 'MICE: multivariate imputation by chained equations', 2019. Available at https://cran.r-project.org/web/packages/mice/, accessed 15 March 2019
[34] Barbiero, A., Ferrari, P.A., Manzi, G.: 'ForImp: imputation of missing values through a forward imputation algorithm', 2015. Available at https://cran.r-project.org/web/packages/ForImp/, accessed 15 March 2019
[35] Stekhoven, D.J.: 'missForest: nonparametric missing value imputation using random forest', 2013. Available at https://cran.r-project.org/web/packages/missForest/, accessed 01 March 2019
[36] van der Loo, M.: 'Simputation: simple imputation', 2017. Available at https://cran.r-project.org/web/packages/simputation/, accessed 05 March 2019
[37] Templ, M., Kowarik, A., Alfons, A., *et al.*: 'VIM: visualization and imputation of missing values', 2019. Available at https://cran.r-project.org/web/packages/VIM/, accessed 09 March 2019
[38] Iskandr: 'Fancyimpute', 2018. Available at https://github.com/iskandr/fancyimpute, accessed 17 March 2019
[39] Cowling, B.J., Freeman, G., Wong, J.Y., *et al.*: 'Spectral regularization algorithms for learning large incomplete matrices', *Eur. Surveillance Bull. Eur. sur les Mal. Transm. = Eur. Commun. Dis. Bull.*, 2013, **18**, (19), p. 20475
[40] Garg, H.: 'A hybrid GSA-GA algorithm for constrained optimization problems', *Inf. Sci. (New York)*, 2019, **478**, pp. 499–523