CBRG: A Novel Algorithm for Handling Missing Data Using Bayesian Ridge Regression and Feature Selection Based on Gain Ratio

SAMIH M. MOSTAFA, ABDELRAHMAN S. ELADIMY, SAFWAT HAMAD, AND HIROFUMI AMANO, (Member, IEEE)

1 Computer Science-Mathematics Department, Faculty of Science, South Valley University, Qena 83523, Egypt
2 Fellow of the Academy of Scientific Research and Technology (ASRT), Cairo 11516, Egypt
3 Scientific Computing Department, Faculty of Computer and Information Sciences, Ain Shams University, Cairo 11566, Egypt
4 Research Institute for Information Technology, Kyushu University, Fukuoka 819-0395, Japan

Corresponding author: Samih M. Mostafa (samih_montser@sci.svu.edu.eg)

This work was supported in part by the Academy of Scientific Research and Technology (ASRT), Egypt, under Grant 6475.

ABSTRACT Existing imputation methods may lead to biased predictions and decrease or increase the statistical influence which leads to improper estimations. Several missing value imputation approaches performance depends on the size of the dataset and the number of missing values within the dataset. In this work, the authors proposed a novel algorithm for manipulating missing data versus some common imputation approaches. The proposed algorithm imputes missing values in cumulative order depending on the gain ratio (GR) feature selection (to select the candidate feature to be manipulated) and the Bayesian Ridge Regression (BRR) technique (to build the predictive model). Each imputed feature will be used to manipulate the missing values in the following selected candidate feature. The proposed algorithm was implemented on eight different datasets after generating different missing values proportions from the missingness mechanisms. The imputation performance was calculated in terms of imputation time, mean absolute error (MAE), coefficient of determination ($R^2$), and root-mean-square error (RMSE). The results show the efficiency of the proposed algorithm when imputing any dataset with any number of missing data from any missingness mechanism.

INDEX TERMS Bayesian ridge regression, imputation, gain ratio, missingness mechanisms, missing value.

I. INTRODUCTION

Data preparation is dealt with as the most significant and time consuming task, which toughly influences the success of the research. The best method for manipulating incomplete instances is to avoid missing data. All skilled researchers still facing missing values that would happen for unknown reasons. Decisions can be made by the researcher at the data collection stage about what data to gather and the method to screen data collecting. The distribution and scale of the variables in the data and the reason for missingness are two important issues to choose the best approach for manipulating missing data [1]. Many imputation algorithms may fail in the imputation of all missing values in the dataset, others give poor performance or consume long imputation time. The proposed algorithm presented in this paper handles these defects by using the most effective features for handling missing values in a cumulative order.

A. FEATURE SELECTION

Feature selection lies in discovering the best subset of possible features from a large set of features. Deleting features that have large amount of missing values in a dataset (e.g., > 50%) is considered a simple solution. However, deleting a feature may result in losing analytical power and ability to observe statistically significant differences and it is frequently a cause of bias, affecting badly on the outcomes. Feature selection requires taking into consideration the missing data mechanism [2].

B. MISSINGNESS MECHANISMS

Manipulating missing data requires detecting the missingness mechanisms (i.e., the cause for existing missing values in a
dataset). This paper deals with the three missingness mechanisms [2]–[5]:

- **Missing completely at random (MCAR)** [6]: Suppose that the missing value indicator matrix \( M = \{M_{ij}\} \) and the complete data \( Y = (y_{ij}) \). The missing data mechanism can be described by the conditional distribution of \( M \) given \( Y \), say \( f(M|Y, \emptyset) \) where \( \emptyset \) symbolizes the unknown parameter. If missingness does not depend on the observed or missing values of the data \( Y \), then
  \[
  f(M|Y, \emptyset) = f(M|\emptyset) \quad \text{for all } Y, \emptyset
  \]

- **Missing at random (MAR)** [6]: Let \( Y_{\text{mis}} \) and \( Y_{\text{obs}} \) symbolize missing data and observed data, respectively. If the missingness do not depend on the data that are missing, but depends only on \( Y_{\text{obs}} \) of \( Y \), then
  \[
  f(M|Y, \emptyset) = f(M|Y_{\text{obs}}, \emptyset) \quad \text{for all } Y_{\text{mis}}, \emptyset
  \]

- **Missing not at random (MNAR)**: If the missingness depends on both missing and observed data [7].

**C. HANDLING MISSING DATA**

The best approach for handling missing data is to avoid it through careful data gathering and follow up, along with determining missing data after the fact (for example, by observing missing data or re-contacting study participants). However, it is generally difficult to avoid missing values in total; therefore, statistical approaches for manipulating missing values are needed. Since missing values are exceptionally complicated, statisticians cannot form a universal set of procedures to handle all cases. Therefore, they run emulation to select the best approach [8]. The methods for manipulating missing values have to custom-made to the proportion of missing values, the size of the dataset and reasons of missingness [2]. The best method to manipulate the missingness is to delete instances that contain missing values. In common, manipulating missing values using deletion leads to accurate estimations just for MCAR [9]. The other method for manipulating missing values and overcomes the disadvantages of the deletion method is the imputation method. Imputation benefits from the complete instances available in the dataset to estimate the missing value [4]. Imputation is regularly implemented before or after a feature selection [2]. The two common model based imputation algorithms are likelihood and regression. The likelihood requires parameter estimation in the existence of missing values, i.e., models and their parameters are assessed using maximum posteriori procedures or maximum likelihood. Imputing missing values using the regression model depends on predicting the unobserved values within a feature depending on the detected values [10]. The BRR technique is used within the proposed algorithm. The proposed algorithm uses a regression model with a regularization parameter. The model satisfies [11]:

\[
y \sim N (\mu, \alpha)
\]

where:

\[
\mu = \beta X = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_p x_p
\]

\[
\beta \sim N \left(0, \lambda^{-1} I_p \right)
\]

\[
\alpha \sim \text{gamma} (\alpha_1, \alpha_2)
\]

\[
\lambda \sim \text{gamma} (\lambda_1, \lambda_2)
\]

where \( y \) is the target feature which is distributed as normal distribution characterized by variance \( \alpha \) and mean \( \mu = \beta X \). \( \beta = \{\beta_0, \beta_1, \beta_2, \ldots, \beta_p\} \) and \( X = \{x_1, x_2, \ldots, x_p\} \) represent the unknown parameters and independent features respectively. \( p \) denotes the number of independent features. \( \lambda \) and \( \alpha \) are regularizing parameters that are distributed as gamma distribution and both are jointly assessed when the model is fitted by maximizing the log marginal likelihood. \( \alpha_1, \alpha_2, \lambda_1 \) and \( \lambda_2 \) are hyper-parameters of the gamma prior distributions.

The structure of this paper as follows: Section 2 shows the literature review of manipulating the missing data. Sections 3 and 4 present the proposed algorithm and explain the experimental implementation, respectively. Results and discussion are described in section 5. Finally, Section 6 concludes this paper.

**II. LITERATURE REVIEW**

Deletion is the preferred method to manipulate missing values. Deleting instances that contain one or more missing values in their feature values is known as list-wise deletion or Complete Case Analysis [4], [12]. Deleting the feature that holds more than a predefined proportion of missing values (e.g., 50%) is known as specific deletion [4], [8], [12], [13]. In addition, deletion can be pair-wise deletion which takes place when statistical procedures cannot use a feature when it contains missing values. Nevertheless, it can benefit from the instance when analyzing other features that do not contain missing values. In the case of each feature in the dataset contains missing data through different instances then using deletion approach may result in losing a big part of the dataset or deletion of the whole dataset [14].

In imputation methods an estimated value is imputed rather than the missed value [12], [13], [15]. The imputed value can be median, mode, mean or any predefined value of the feature that holds missing value [16]–[18], or can be acquired from case substitution. Imputed value can also be estimated using KNN (K Nearest Neighbors) [3], cold deck imputation [19], expectation-maximization imputation [20]–[22], hot-deck imputation [23], etc. In techniques contain prediction models, a model is built based on the available information within the dataset, then this model is used to predict missing values [24]. Imputation methods are used to manipulate missing values if; i) the missing values are MCAR or MAR type, ii) the feature that contains missing values has statistical influence with the target feature, iii) deletion of instances decreases the size of the dataset, which in turn affects on building the predictive model, iv) an instance does not include missing values across many features [14]. For MCAR missingness mechanism type can be manipulated using maximum likelihood methods or listwise deletion. There are no general approaches to manipulate missing values of MNAR missingness mechanism type [4], [25].
Imputation types are single imputation or multiple imputation. In single imputation, a particular value is imputed rather than the missed value [19]. Multiple imputation, where \( m \) complete datasets are generated by imputing the missing values \( m \) times, the final imputed dataset is the analysis average of these \( m \) datasets [26]. Although multiple imputation prerequisites more resources [27], it has benefits more than other methods such as single imputation, maximum likelihood techniques and deletion [28]. Inverse Probability Weighting (IPW) approaches are other methods to handle missing values. IPW depends on the inverse of the detected probability to weight detected instances by representing the entire data even the missing values. However, the performance of imputation method is better [4], [29]. Singular value decomposition (SVD), partial least squares (PLS) and ordinary least squares (OLS) are also good choices for multiple imputation [4]. Manipulating missing values in datasets containing ordinal and binary features using Multi Variate Normal Imputation (MVNI) and Fully Conditional Specification (FCS) approaches generate similar and less biased results. The model specification is granted easily by MVNI, but as a result of its unrealistic nature assumption, some people may found a problem with it. FCS tends to have a difficult model requirements as a result of demanding a single regression model for every variable whose missing values will be imputed [30]. KNN is used by FINNIM which is a practical nonparametric iterative multiple imputation approach to estimate missing values [31]. Successive regression trees are used as a multiple imputation conditional model, which has the ability to identify complex relations and need small improvement by the user [32]. Manipulating missing values with predictive mean matching (PMM) approach is implemented using a randomly drawn instance from a set of detected instances whose predictive mean is close to the predictive mean of missing values [4]. Like PMM, local residual draw (LRD) approach manipulates missing values using the predictive mean, but with extra randomly drawn from the residuals set of detected instances with predictive means close to that of the missing value [33]. Reinforcement programming (RP) which depends on Reinforcement Learning gives better performance over mean per category imputation, zero imputation and genetic algorithm (GA) in terms of the sum of square error and computational time [34]. Cumulative linear regression which depends on the linear regression technique to manipulate missing data works better with large and small datasets [3]. Manipulating missing data in variables of interest with the aid of observed values from other variables depends on the similarities of the observation values within the donor variable to manipulate the missing data within the recipient variable. It works well when the mount of the missing values is large [6].

III. PROPOSED ALGORITHM

This section exhibits the proposed algorithm in detail. The following procedural steps help in elaborating the proposed algorithm described in Fig. 1:

**Step 1:** Suppose that the proposed algorithm takes a dataset \( D \) which holds missing values as input, then by partitioning \( D \) into two sets. The first set \( X^{(comp)} \) contains all complete features and the second set \( X^{(mis)} \) contains all incomplete features. The authors assume that the output feature contains no missing values, so \( X^{(comp)} \) consists of all complete features plus the output feature \( y \).

**Step 2:** The proposed algorithm we called Cumulative Bayesian Ridge with Gain ratio (CBRG), uses the GR feature selection. The choice of the candidate feature to be imputed must offer the highest GR with the \( y \) (Algorithm 1). The GR criterion defined by Quinlan is given by (4) [35]:

\[
IG(A_k, X) = I(X) - E(A_k, X) \quad (2)
\]

\[
IV(A_k) = - \frac{1}{n} \sum_{i=1}^{n} \frac{|X_i|}{|X|} \log_2 \left( \frac{|X_i|}{|X|} \right) \quad (3)
\]

\[
GR = \frac{IG(A_k, X)}{IV(A_k)} \quad (4)
\]

where \( IG \) is the information gain, \( X \) is the set of cases, \( I \) specifies the entropy, and \( E \) is the expected information of the feature \( A_k \), \( n \) is the number of possible values of feature \( A_k \), and \( IV \) is the intrinsic value.

**Step 3:** After choosing the candidate feature \( X^{(mis)}_g \), the model is fitted using the cumulative formula given by (5) with \( X^{(comp)} \) as the independent feature and the candidate feature as dependent. The chosen feature removed from \( X^{(mis)} \) and after imputing missing values within the \( X^{(mis)}_g \) the imputed feature \( X^{(mis)}_{imp} \), is added to \( X^{(comp)} \). Now \( X^{(comp)} \) includes all
Algorithm 1 CBRG

Input:
- $D$: A dataset with missing values containing $n$ instances.

Output:
- $D_{\text{imputed}}$: A dataset with all missing features imputed.

Definitions:
- $X^{(\text{comp})}$: set of complete features.
- $X^{(\text{mis})}$: set of incomplete features.
- $X^{(\text{mis})}_i$: imputed feature from $X^{(\text{mis})}_i$.
- $m$: number of features containing missing values.
- $GR \left( X^{(\text{mis})}_i, y \right)_{\alpha, \lambda}$: GR between $X^{(\text{mis})}_i$ and $y$, $l \in [1, \ldots, m]$.

Begin

1. Split $D$ into $X^{(\text{comp})}$ and $X^{(\text{mis})}$.
2. From $X^{(\text{mis})}$ select $X^{(\text{mis})}_j$ that satisfies the condition:
   $$\max \left( GR \left( X^{(\text{mis})}_j, y \right) \right).$$
3. While $X^{(\text{mis})} = \emptyset$
   i. $g \leftarrow$ index of the candidate feature from $X^{(\text{mis})}$.
   ii. Fit the BRR model on $X^{(\text{comp})}$ as independent features and $X^{(\text{mis})}_g$ as dependent feature using the cumulative formula given by (5).
   iii. $X^{(\text{mis})}_i \leftarrow$ Impute the missing data in $X^{(\text{mis})}_g$ with the fitted model.
   iv. Delete $X^{(\text{mis})}_g$ from $X^{(\text{mis})}$ and add $X^{(\text{mis})}_i$ to $X^{(\text{comp})}$.
4. End While

End

Step 4: Repeat from step 2 until $X^{(\text{mis})}$ is empty, at that point return $X^{(\text{comp})}$ as the imputed dataset.

**IV. EXPERIMENTAL IMPLEMENTATION**

**A. DATASETS**

The datasets were gathered from several databases repository and differ from each other in size and type. The authors used eight different datasets that usually studied in the literature) Table 1. The BNG_heart_statlog and Poker Hand datasets are considered massive datasets, therefore the analysis was implemented on randomly sampled sub-datasets of 10000, 15000, 20000, and 50000 of instances from both of them [4].

In every dataset, the proportions of missing values were generated from each missingness mechanism using ampute function from the R environment [44].

**B. PERFORMANCE EVALUATION**

The performance evaluation was calculated using RMSE, MAE, $R^2$ and time of imputation in seconds [3].

$$\text{RMSE} = \sqrt{\frac{\sum_{i=1}^{n} (y_i - \hat{y}_i)^2}{n}} \quad (6)$$

| Dataset name                  | Instances | Features | MR    |
|------------------------------|-----------|----------|-------|
| Diabetes [36]                | 442       | 11       | 10% - 40% |
| graduate admissions [37]     | 500       | 8        | 10% - 40% |
| profit estimation of companies [38] | 1,000    | 6        | 10% - 40% |
| red & white wine dataset [39] | 4,898     | 12       | 10% - 40% |
| California [40]              | 20,640    | 9        | 10% - 40% |
| Diamonds [41]                | 53,940    | 10       | 10% - 40% |
| BNG_heart_statlog [42]       | 1,000,000 | 14       | 10% - 40% |
| Poker Hand dataset [43]      | 1,025,010 | 11       | 10% - 40% |

**TABLE 2.** The algorithms used in the comparison.

| Package (method name) | Description |
|-----------------------|-------------|
| impute (mice) [44], [45] | imputes using the multivariate imputation by chained equations algorithm. |
| autoimpute (LeastSquares) [46] | generates estimations using the least squares methodology. |
| autoimpute (Norm) [46] | produces a Gaussian distribution using the sample variance and mean of the observed data. |
| autoimpute (Stochastic) [46] | samples from the regression’s error distribution and adds the random draw to the estimation. |
| impute (Fast KNN) [47] | implements K-Dimensional tree to find k nearest neighbor and manipulates missing values using the weighted average of them. |
| impute (EMI) [47] | handles missing values using Expectation Maximization Imputation. |
where $\hat{y}_l$ and $y_l$ are the predicted and real values of the $l$th instances, respectively. $\bar{y}$ is the mean of the observed data. $n$ is the number of instances.

The CBRG was compared versus six comparison algorithms presented briefly in Table 2. Multiple Imputation by Chained Equations (MICE) is one of the best methods for manipulating missing data problems. MICE assumes that the missing data are of MAR type. It manipulates missing data in a feature using an imputation model per feature. By default, linear regression and Logistic regression are used to impute continuous and categorical missing values respectively. In MICE, predictive models are used to manipulate missing data in an iterative series. When convergence occurs, these iterations should be stopped. Least squares methodology is used by the LeastSquares method, which is a standard method in the regression analysis. It minimizes the sum of squared residuals (i.e., the difference between the imputed value predicted by a predictive model and the observed value). When the data are of MCAR type, coefficients of leastSquares are stable (i.e., unbiased when the sample dimension increases) but they are not fully efficient. Assessing models using weighted least squares leads to better results. The Norm method creates a Gaussian distribution using the sample mean and variance of the observed data. Randomly samples from this distribution are used to manipulate missing data by Norm method. In the Stochastic method, missing values are manipulated using the least squares methodology. It adds the random draw to the prediction after creating samples from the regression’s error distribution. Although this method can be used directly, such behavior is not preferred. The Fast KNN method manipulates missing data with a passed in initial impute fn (mean impute) and then uses the resulting initial imputed dataset to build
a KDTree which will be used to assess nearest neighbours. The 'k' nearest neighbours will be taken as a weighted average.

The experiments were implemented using a computer with the following specification: 4 GB memory, Intel Core i5-2400 (3.10 GHz) processor, 500 GB HDD, Windows 10 OS, and R (version 3.5.2) and Python (version 3.7) programming language.

V. RESULTS, DISCUSSION AND LIMITATIONS OF THE PROPOSED METHOD

Figs. 2 to 5 show the performance improvement of the proposed algorithm versus the compared algorithms, which was evaluated using the four evaluation metrics. Tables 3 to 16 exhibit the performance evaluation of the CBRG versus the compared algorithms for each dataset in detail. The figures show that the performance varies depending on the missingness mechanisms, the dimension of the dataset and the number of missing values in the dataset. The computational complexity of the proposed algorithm is $O(n)$.

This section is subdivided into three subsections. The first subsection presents the accuracy analysis, the second explains the error analysis, and the third subsection presents the imputation time. Finally, the fourth subsection presents limitations of the proposed algorithm.

A. ACCURACY ANALYSIS

This subsection exhibits that CBRG shows better accuracy in most cases. In what follows, the accuracy analysis is discussed in details. The accuracy is known as how well the model in predicting the unknown data. Fig. 2 shows the improvement percentage of $R^2$ defined by (8) for CBRG versus the compared algorithms. When CBRG is compared with Stochastic, Norm, Fast KNN, Expectation Maximization Imputation (EMI) in all missingness mechanisms, the improvement of $R^2$ in all datasets for the CBRG is better than Stochastic, Norm, Fast KNN and EMI, but in MNAR, CBRG is worse than Stochastic when implemented on the California dataset. When CBRG is compared with Least-Squares in MAR, $R^2$ of CBRG is better than LeastSquares when implemented on the profit estimation of companies and...
diamonds datasets. $R^2$ of CBRG equals to LeastSquares in all generated samples from BNG_heart_statlog and the Poker Hand datasets, and $R^2$ of CBRG is worse than LeastSquares when implemented on the rest of the datasets. In MCAR, $R^2$ of CBRG is equal to LeastSquares when implemented on all generated samples from BNG_heart_statlog and the Poker Hand datasets, and worse when implemented on the rest of the datasets. In MNAR, $R^2$ of CBRG is better than LeastSquares when implemented on the profit estimation of companies dataset. $R^2$ of CBRG is equal to LeastSquares when implemented on all generated samples from BNG_heart_statlog and the Poker Hand datasets, and worse when implemented on the rest of the datasets. In MCAR, it was observed that $R^2$ of CBRG is equal to MICE when implemented on diamonds and all generated samples from BNG_heart_statlog and the Poker Hand datasets, and worse when implemented on the rest of the datasets. When CBRG is compared with Multivariate Imputation by Chained Equations (MICE) in MAR, it was observed that $R^2$ of CBRG is equal to MICE when implemented on diamonds and all generated samples from BNG_heart_statlog and the Poker Hand datasets, and worse when implemented on the rest of the datasets. In MCAR, it was observed that $R^2$ of CBRG is equal to MICE when implemented on all generated samples from BNG_heart_statlog and the Poker Hand datasets, and worse when implemented on the rest of the datasets. In MNAR, it was observed that $R^2$ of CBRG is equal to MICE when implemented on diamonds, BNG (10000), Poker (10000), Poker (15000), Poker (20000) and Poker (50000) datasets, and worse when implemented on the rest of the datasets.

Selecting large samples from a dataset leads the distribution of the samples to be very close to a normal distribution. Since the CBRG depends on the BRR technique which in turn assumes that the independent features have a normal distribution, hence CBRG gives good accuracy when implemented on the generated samples datasets.

### B. ERROR ANALYSIS

This subsection exhibits that CBRG shows lower errors in most cases. In what follows, the error analysis is discussed in details. The error analysis was implemented by calculating MAE given by (7) and RMSE defined by (6). Both Fig. 3 and Fig. 4 show the improvement percentage of MAE and RMSE for the proposed algorithm versus the compared algorithms, respectively. When CBRG is compared with Norm in all missingness mechanisms, it was observed that CBRG granted lower error than Norm when implemented on all datasets. When CBRG is compared with Stochastic in all missingness mechanisms, MAE of CBRG is better than Stochastic when implemented on all datasets except the diamonds dataset. In MAR, the RMSE of CBRG is worse than Stochastic when implemented on the diamonds dataset. In MNAR, the RMSE of CBRG is worse than Stochastic when implemented on diamond and diabetes datasets. In MCAR, RMSE of CBRG is better than Stochastic when implemented on all datasets. When CBRG is compared with Fast KNN in MAR and MNAR, MAE of CBRG is better than Fast KNN when implemented on all datasets. In MCAR, MAE of CBRG is better than Fast KNN when implemented on all datasets except the California dataset. In MNAR, RMSE of CBRG is better than Fast KNN when implemented on all datasets. In MAR, RMSE of CBRG is better than Fast KNN when implemented on all datasets except the California and Poker (50000) datasets. In MCAR, CBRG is better than Fast KNN when implemented on all datasets except in California, BNG (15000), Poker (10000), Poker (15000), and Poker (20000) datasets. When CBRG is compared with EMI, in all missingness mechanisms, MAE of CBRG is better than EMI when implemented on all datasets. In MNAR, RMSE of CBRG is better than EMI when implemented on all datasets except the California and Poker (20000) datasets. In MCAR, RMSE of CBRG is better than EMI when implemented on all datasets except in California, BNG (15000), Poker (10000), Poker (15000), and Poker (20000) datasets. When CBRG is compared with LeastSquares in MAR, MAE of CBRG is better than LeastSquares when implemented on all datasets except diabetes, red & white wine, diamonds, BNG (15000), BNG (20000) and BNG (50000) datasets. In MAR, MAE of CBRG is better than LeastSquares when implemented on all datasets except diabetes, red & white wine, diamonds, BNG (15000), BNG (20000) and BNG (50000) datasets. In MAR, RMSE of CBRG is better than LeastSquares when implemented on all datasets except diabetes, red & white wine, California, diamonds, BNG (10000), BNG (15000), BNG (20000) and BNG (50000) datasets. In MAR, RMSE of CBRG is better than LeastSquares when implemented on all datasets except diabetes, red & white wine, California, diamonds, BNG (15000), BNG (20000) and BNG (50000) datasets. When CBRG is compared with MICE, in MAR, MAE of CBRG is better than MICE when implemented on graduate admissions, profit estimation of companies and Poker (10000) datasets, and worse when implemented on the rest of the datasets. In MCAR, MAE of CBRG is better than MICE when implemented on graduate admissions, profit estimation of companies and Poker (10000) and Poker (50000) datasets, and worse when implemented on the rest of the datasets.
### TABLE 3. \( R^2 \), imputation time, MAE, and RMSE of the CBRG versus the compared algorithms (graduate admissions dataset).

| Grade       | Methods       | \% Missing Value | \( R^2 \) | Time | MAE  | RMSE  |
|-------------|---------------|------------------|---------|------|------|-------|
| Graduate    | CBRG          | 10               | 0.972   | 0.9572 | 0.9552 | 0.9760 | 0.9763 |
|             |               | 20               | 0.9893 | 0.9787 | 0.9796 | 0.9796 | 0.9806 |
| Competitors |               |                  | 0.9796 | 0.9797 | 0.9796 | 0.9794 | 0.9794 |
|             |               |                  | 0.9796 | 0.9797 | 0.9796 | 0.9794 | 0.9794 |
|             |               |                  | 0.9796 | 0.9797 | 0.9796 | 0.9794 | 0.9794 |
| Average     |               |                  | 0.9796 | 0.9797 | 0.9796 | 0.9794 | 0.9794 |
| Improvement |               |                  | -0.0003 | 0.0000 | 0.0000 | 0.0000 | 0.0000 |
| MCGR        | CBRG          | 10               | 0.972   | 0.9572 | 0.9552 | 0.9760 | 0.9763 |
|             |               | 20               | 0.9893 | 0.9787 | 0.9796 | 0.9796 | 0.9806 |
| Competitors |               |                  | 0.9796 | 0.9797 | 0.9796 | 0.9794 | 0.9794 |
|             |               |                  | 0.9796 | 0.9797 | 0.9796 | 0.9794 | 0.9794 |
|             |               |                  | 0.9796 | 0.9797 | 0.9796 | 0.9794 | 0.9794 |
| Average     |               |                  | 0.9796 | 0.9797 | 0.9796 | 0.9794 | 0.9794 |
| Improvement |               |                  | -0.0003 | 0.0000 | 0.0000 | 0.0000 | 0.0000 |

### TABLE 4. \( R^2 \), imputation time, MAE, and RMSE of the CBRG versus the compared algorithms (Diabetes dataset).

| Grade       | Methods       | \% Missing Value | \( R^2 \) | Time | MAE  | RMSE  |
|-------------|---------------|------------------|---------|------|------|-------|
| Graduate    | CBRG          | 10               | 0.972   | 0.9572 | 0.9552 | 0.9760 | 0.9763 |
|             |               | 20               | 0.9893 | 0.9787 | 0.9796 | 0.9796 | 0.9806 |
| Competitors |               |                  | 0.9796 | 0.9797 | 0.9796 | 0.9794 | 0.9794 |
|             |               |                  | 0.9796 | 0.9797 | 0.9796 | 0.9794 | 0.9794 |
|             |               |                  | 0.9796 | 0.9797 | 0.9796 | 0.9794 | 0.9794 |
| Average     |               |                  | 0.9796 | 0.9797 | 0.9796 | 0.9794 | 0.9794 |
| Improvement |               |                  | -0.0003 | 0.0000 | 0.0000 | 0.0000 | 0.0000 |
| MCGR        | CBRG          | 10               | 0.972   | 0.9572 | 0.9552 | 0.9760 | 0.9763 |
|             |               | 20               | 0.9893 | 0.9787 | 0.9796 | 0.9796 | 0.9806 |
| Competitors |               |                  | 0.9796 | 0.9797 | 0.9796 | 0.9794 | 0.9794 |
|             |               |                  | 0.9796 | 0.9797 | 0.9796 | 0.9794 | 0.9794 |
|             |               |                  | 0.9796 | 0.9797 | 0.9796 | 0.9794 | 0.9794 |
| Average     |               |                  | 0.9796 | 0.9797 | 0.9796 | 0.9794 | 0.9794 |
| Improvement |               |                  | -0.0003 | 0.0000 | 0.0000 | 0.0000 | 0.0000 |

### TABLE 5. \( R^2 \), imputation time, MAE, and RMSE of the CBRG versus the compared algorithms (Graduate Admissions dataset).

| Grade       | Methods       | \% Missing Value | \( R^2 \) | Time | MAE  | RMSE  |
|-------------|---------------|------------------|---------|------|------|-------|
| Graduate    | CBRG          | 10               | 0.972   | 0.9572 | 0.9552 | 0.9760 | 0.9763 |
|             |               | 20               | 0.9893 | 0.9787 | 0.9796 | 0.9796 | 0.9806 |
| Competitors |               |                  | 0.9796 | 0.9797 | 0.9796 | 0.9794 | 0.9794 |
|             |               |                  | 0.9796 | 0.9797 | 0.9796 | 0.9794 | 0.9794 |
|             |               |                  | 0.9796 | 0.9797 | 0.9796 | 0.9794 | 0.9794 |
| Average     |               |                  | 0.9796 | 0.9797 | 0.9796 | 0.9794 | 0.9794 |
| Improvement |               |                  | -0.0003 | 0.0000 | 0.0000 | 0.0000 | 0.0000 |
| MCGR        | CBRG          | 10               | 0.972   | 0.9572 | 0.9552 | 0.9760 | 0.9763 |
|             |               | 20               | 0.9893 | 0.9787 | 0.9796 | 0.9796 | 0.9806 |
| Competitors |               |                  | 0.9796 | 0.9797 | 0.9796 | 0.9794 | 0.9794 |
|             |               |                  | 0.9796 | 0.9797 | 0.9796 | 0.9794 | 0.9794 |
|             |               |                  | 0.9796 | 0.9797 | 0.9796 | 0.9794 | 0.9794 |
| Average     |               |                  | 0.9796 | 0.9797 | 0.9796 | 0.9794 | 0.9794 |
| Improvement |               |                  | -0.0003 | 0.0000 | 0.0000 | 0.0000 | 0.0000 |
TABLE 5. R², imputation time, MAE, and RMSE of the CBRG versus the compared algorithms (Profit dataset).

| Method   | CBRG | Local Squares | Stochastic | Norm | MICE | FastKNN | ENN | EMI |
|----------|------|---------------|------------|------|------|---------|-----|-----|
| % missing value | $\omega$ | Time | $\omega$ | Time | $\omega$ | Time | $\omega$ | Time | $\omega$ | Time | $\omega$ | Time | $\omega$ | Time | $\omega$ | Time |
| 10        | 0.8714 | 1.2674 | 0.7621 | 1.1511 | 0.7831 | 1.1319 | 0.8309 | 1.0228 | 0.9772 | 0.9408 | 0.8972 | 0.8632 | 0.8630 | 0.8630 | 0.8630 | 0.8630 |
| 20        | 0.8655 | 1.2759 | 0.7593 | 1.1500 | 0.7863 | 1.1315 | 0.8364 | 1.0296 | 0.9817 | 0.9472 | 0.8972 | 0.8632 | 0.8630 | 0.8630 | 0.8630 | 0.8630 |
| 30        | 0.8678 | 1.2995 | 0.7873 | 1.1282 | 0.7516 | 1.1401 | 0.8004 | 1.0115 | 0.9785 | 1.2592 | 0.9611 | 0.8216 | 0.8213 | 0.8213 | 0.8213 | 0.8213 |
| Average   | 0.8681 | 1.2985 | 0.7566 | 1.1027 | 0.7512 | 1.1037 | 0.8177 | 0.9797 | 0.9524 | 0.9356 | 0.9210 | 0.8164 | 0.8150 | 0.8150 | 0.8150 | 0.8150 |

TABLE 6. R², imputation time, MAE, and RMSE of the CBRG versus the compared algorithms (Wine dataset).

| Method   | CBRG | Local Squares | Stochastic | Norm | MICE | FastKNN | ENN | EMI |
|----------|------|---------------|------------|------|------|---------|-----|-----|
| % missing value | $\omega$ | Time | $\omega$ | Time | $\omega$ | Time | $\omega$ | Time | $\omega$ | Time | $\omega$ | Time | $\omega$ | Time | $\omega$ | Time |
| 10        | 0.9005 | 1.2172 | 0.9910 | 1.0215 | 0.8956 | 1.0122 | 0.9517 | 0.9841 | 0.9914 | 0.9660 | 0.9106 | 0.9059 | 0.8770 | 0.8770 | 0.8770 | 0.8770 |
| 20        | 0.8721 | 1.2186 | 0.9795 | 1.0225 | 0.8460 | 1.0647 | 0.9195 | 0.9904 | 0.9816 | 0.9647 | 0.9011 | 0.8625 | 0.8625 | 0.8625 | 0.8625 | 0.8625 |
| 30        | 0.8701 | 1.2195 | 0.9787 | 1.0167 | 0.8406 | 1.0647 | 0.9195 | 0.9904 | 0.9816 | 0.9647 | 0.9011 | 0.8625 | 0.8625 | 0.8625 | 0.8625 | 0.8625 |

TABLE 5. R², imputation time, MAE, and RMSE of the CBRG versus the compared algorithms (Profit dataset).

| Method   | CBRG | Local Squares | Stochastic | Norm | MICE | FastKNN | ENN | EMI |
|----------|------|---------------|------------|------|------|---------|-----|-----|
| % missing value | $\omega$ | Time | $\omega$ | Time | $\omega$ | Time | $\omega$ | Time | $\omega$ | Time | $\omega$ | Time | $\omega$ | Time | $\omega$ | Time |
| 10        | 0.9122 | 1.0224 | 0.9616 | 1.1155 | 0.9566 | 1.1178 | 0.9609 | 1.0599 | 0.9589 | 0.9283 | 0.9011 | 0.8789 | 0.8789 | 0.8789 | 0.8789 | 0.8789 |
| 20        | 0.9457 | 1.2445 | 0.9655 | 1.0625 | 0.9231 | 1.2116 | 0.9210 | 1.0410 | 0.8764 | 0.7828 | 0.9029 | 0.8789 | 0.8789 | 0.8789 | 0.8789 | 0.8789 |
| 30        | 0.9053 | 1.2090 | 0.9602 | 1.1285 | 0.9494 | 1.1224 | 0.9695 | 1.0514 | 0.9891 | 0.8428 | 0.7009 | 0.8258 | 0.8258 | 0.8258 | 0.8258 | 0.8258 |

TABLE 6. R², imputation time, MAE, and RMSE of the CBRG versus the compared algorithms (Wine dataset).

| Method   | CBRG | Local Squares | Stochastic | Norm | MICE | FastKNN | ENN | EMI |
|----------|------|---------------|------------|------|------|---------|-----|-----|
| % missing value | $\omega$ | Time | $\omega$ | Time | $\omega$ | Time | $\omega$ | Time | $\omega$ | Time | $\omega$ | Time | $\omega$ | Time | $\omega$ | Time |
| 10        | 0.9169 | 1.1018 | 0.9769 | 1.1285 | 0.8553 | 1.1137 | 0.9086 | 1.0134 | 0.9892 | 0.9716 | 0.9237 | 0.8672 | 0.8672 | 0.8672 | 0.8672 | 0.8672 |
| 20        | 0.9367 | 1.0981 | 0.9763 | 1.1105 | 0.8406 | 1.0722 | 0.8901 | 1.0116 | 0.9892 | 0.9716 | 0.9237 | 0.8672 | 0.8672 | 0.8672 | 0.8672 | 0.8672 |
| 30        | 0.9308 | 1.0971 | 0.9714 | 1.1125 | 0.8331 | 1.0740 | 0.8850 | 1.0075 | 0.9892 | 0.9716 | 0.9237 | 0.8672 | 0.8672 | 0.8672 | 0.8672 | 0.8672 |
### Table 7. R², imputation time, MAE, and RMSE of CBRG versus the compared algorithms (California dataset).

| Algorithm | Table 7 | R² | imputation time | MAE | RMSE |
|-----------|---------|----|----------------|-----|------|
| CBRG      | 0.09272 | 10 | 0.09789 | 0.09478 | 0.10597 | 0.11769 |
| MICE      | 0.09272 | 10 | 0.09789 | 0.09478 | 0.10597 | 0.11769 |
| FMM        | 0.09272 | 10 | 0.09789 | 0.09478 | 0.10597 | 0.11769 |
| EM        | 0.09272 | 10 | 0.09789 | 0.09478 | 0.10597 | 0.11769 |

### Table 8. R², imputation time, MAE, and RMSE of CBRG versus the compared algorithms (diamond dataset).

| Algorithm | Table 8 | R² | imputation time | MAE | RMSE |
|-----------|---------|----|----------------|-----|------|
| CBRG      | 0.09272 | 10 | 0.09789 | 0.09478 | 0.10597 | 0.11769 |
| MICE      | 0.09272 | 10 | 0.09789 | 0.09478 | 0.10597 | 0.11769 |
| FMM        | 0.09272 | 10 | 0.09789 | 0.09478 | 0.10597 | 0.11769 |
| EM        | 0.09272 | 10 | 0.09789 | 0.09478 | 0.10597 | 0.11769 |
TABLE 9. R², imputation time, MAE, and RMSE of CBRG versus the compared algorithms (sample dataset from BNG_heart_statlog dataset with size (10000)).

| R² | imputation time | MAE | RMSE |
|----------------|----------------|-----|------|
| CBRG | Least Squares | Stochastic | Norm | MICE | Fast KNN | EMI |
| -------|----------------|----------------|-----|------|----------|------|
| 0.991 | 0.981 | 0.971 | 0.966 | 0.965 | 0.964 | 0.963 |
| 0.983 | 0.978 | 0.965 | 0.961 | 0.960 | 0.959 | 0.958 |
| 0.968 | 0.961 | 0.952 | 0.948 | 0.947 | 0.945 | 0.944 |
| 0.960 | 0.952 | 0.944 | 0.941 | 0.939 | 0.937 | 0.935 |

Average | 0.980 | 0.972 | 0.966 | 0.963 | 0.960 | 0.958 | 0.956 |

**TABLE 10.** R², imputation time, MAE, and RMSE of CBRG versus the compared algorithms (sample dataset from BNG_heart_statlog dataset with size (15000)).

| R² | imputation time | MAE | RMSE |
|----------------|----------------|-----|------|
| CBRG | Least Squares | Stochastic | Norm | MICE | Fast KNN | EMI |
| -------|----------------|----------------|-----|------|----------|------|
| 0.991 | 0.981 | 0.971 | 0.966 | 0.965 | 0.964 | 0.963 |
| 0.983 | 0.978 | 0.965 | 0.961 | 0.960 | 0.959 | 0.958 |
| 0.968 | 0.961 | 0.952 | 0.948 | 0.947 | 0.945 | 0.944 |
| 0.960 | 0.952 | 0.944 | 0.941 | 0.939 | 0.937 | 0.935 |

Average | 0.980 | 0.972 | 0.966 | 0.963 | 0.960 | 0.958 | 0.956 |
### TABLE 11. \( R^2 \), imputation time, MAE, and RMSE of CBRG versus the compared algorithms (sample dataset from BNG_heart_statlog dataset with size (20000)).

| Method | Imputation time | MAE | RMSE |
|--------|----------------|-----|------|
| CBRG   | 2.098          | 0.472 | 0.983 |
| Least Squares | 2.382 | 0.982 | 0.972 |
| Stochastic | 2.382 | 0.982 | 0.972 |
| Norm | 2.382 | 0.982 | 0.972 |

### TABLE 12. \( R^2 \), imputation time, MAE, and RMSE of CBRG versus the compared algorithms (sample dataset from Poker Hand dataset with size (50000)).

| Method | Imputation time | MAE | RMSE |
|--------|----------------|-----|------|
| CBRG   | 2.098          | 0.472 | 0.983 |
| Least Squares | 2.382 | 0.982 | 0.972 |
| Stochastic | 2.382 | 0.982 | 0.972 |
| Norm | 2.382 | 0.982 | 0.972 |

---

S. M. Mostafa et al.: CBRG: A Novel Algorithm for Handling Missing Data

IEEE Access

VOLUME 8, 2020
### TABLE 13. $R^2$, imputation time, MAE, and RMSE of CBRG versus the compared algorithms (sample dataset from Poker Hand dataset with size (10000)).

| CBRG | Least Squares | Stochastic | Norm | MICE | Fast KNN | EMI |
|------|---------------|------------|------|------|----------|-----|
| % missing value | $R^2$ | Score | Time | $R^2$ | Score | Time | $R^2$ | Score | Time | $R^2$ | Score | Time | $R^2$ | Score | Time | $R^2$ | Score |
| 10   | 0.9800 | 0.5675 | 0.9900 | 1.0194 | 0.9795 | 0.9399 | 0.9786 | 0.7185 | 0.9000 | 15.4999 | 0.0684 | 12.1584 | 0.9615 | 4.0877 |
| 20   | 0.9785 | 0.3847 | 0.9672 | 1.7480 | 0.9756 | 0.9714 | 0.9569 | 0.7877 | 0.9000 | 19.5674 | 0.0647 | 16.2557 | 0.9643 | 6.5904 |
| 30   | 0.9690 | 0.4372 | 0.9648 | 1.8481 | 0.9690 | 0.9231 | 0.9408 | 0.6465 | 0.9000 | 28.4135 | 0.0565 | 38.2132 | 0.9462 | 12.5210 |
| 40   | 0.9593 | 0.3899 | 0.9585 | 2.0023 | 0.9592 | 0.8905 | 0.8872 | 0.5975 | 0.9000 | 32.8217 | 0.0485 | 23.3217 | 0.9582 | 16.2954 |
| 50   | 0.9483 | 0.3920 | 0.9379 | 2.1376 | 0.9490 | 0.8492 | 0.7820 | 0.5243 | 0.9000 | 36.8160 | 0.0405 | 28.4135 | 0.9538 | 19.9961 |

### TABLE 14. $R^2$, imputation time, MAE, and RMSE of CBRG versus the compared algorithms (sample dataset from Poker Hand dataset with size (15000)).

| CBRG | Least Squares | Stochastic | Norm | MICE | Fast KNN | EMI |
|------|---------------|------------|------|------|----------|-----|
| % missing value | $R^2$ | Score | Time | $R^2$ | Score | Time | $R^2$ | Score | Time | $R^2$ | Score | Time | $R^2$ | Score | Time | $R^2$ | Score |
| 10   | 0.9875 | 0.3586 | 0.9779 | 1.3169 | 0.9789 | 0.9393 | 0.9516 | 0.1695 | 0.9000 | 15.5827 | 0.0643 | 18.2311 | 0.9738 | 4.1111 |
| 20   | 0.9850 | 0.4087 | 0.9677 | 1.7480 | 0.9740 | 0.9726 | 0.9569 | 0.7877 | 0.9000 | 20.0996 | 0.0625 | 22.6873 | 0.9643 | 6.5904 |
| 30   | 0.9755 | 0.4367 | 0.9648 | 1.8481 | 0.9678 | 0.9231 | 0.9408 | 0.6465 | 0.9000 | 29.7506 | 0.0547 | 33.8451 | 0.9582 | 16.2954 |
| 40   | 0.9650 | 0.4467 | 0.9585 | 2.0023 | 0.9592 | 0.8905 | 0.8872 | 0.5975 | 0.9000 | 33.7337 | 0.0467 | 27.2312 | 0.9538 | 19.9961 |
| 50   | 0.9545 | 0.3890 | 0.9379 | 2.1376 | 0.9490 | 0.8492 | 0.7820 | 0.5243 | 0.9000 | 37.7133 | 0.0387 | 31.2817 | 0.9538 | 19.9961 |
TABLE 15. $R^2$, imputation time, MAE, and RMSE of CBRG versus the compared algorithms (sample dataset from Poker Hand dataset with size (20000)).

|          | Poker (20000) dataset |          |          |          |          |
|----------|-----------------------|----------|----------|----------|----------|
|          | CBRG                  | LeastSquares | Stochastic | Norm     | MICE     | Fast KNN | EMI       |
| % missing value | $R^2$ | Imputation time | MAE | RMSE | $R^2$ | Imputation time | MAE | RMSE | $R^2$ | Imputation time | MAE | RMSE | $R^2$ | Imputation time | MAE | RMSE | $R^2$ | Imputation time | MAE | RMSE |
| 10       | 0.9908                | 0.0051    | 0.0051   | 0.0051   | 0.0051   | 0.0051   | 0.0051   | 0.0051   | 0.0051   | 0.0051   | 0.0051   | 0.0051   | 0.0051   | 0.0051   | 0.0051   | 0.0051   | 0.0051   | 0.0051   |
| 20       | 0.9908                | 0.0051    | 0.0051   | 0.0051   | 0.0051   | 0.0051   | 0.0051   | 0.0051   | 0.0051   | 0.0051   | 0.0051   | 0.0051   | 0.0051   | 0.0051   | 0.0051   | 0.0051   | 0.0051   | 0.0051   |
| 30       | 0.9908                | 0.0051    | 0.0051   | 0.0051   | 0.0051   | 0.0051   | 0.0051   | 0.0051   | 0.0051   | 0.0051   | 0.0051   | 0.0051   | 0.0051   | 0.0051   | 0.0051   | 0.0051   | 0.0051   | 0.0051   |
| 40       | 0.9908                | 0.0051    | 0.0051   | 0.0051   | 0.0051   | 0.0051   | 0.0051   | 0.0051   | 0.0051   | 0.0051   | 0.0051   | 0.0051   | 0.0051   | 0.0051   | 0.0051   | 0.0051   | 0.0051   | 0.0051   |

C. IMPUTATION TIME

This subsection exhibits that CBRG shows better imputation time in most cases. In what follows, the imputation time analysis is discussed in details. Fig. 5 shows the improvement percentage of imputation time for the proposed algorithm versus the compared algorithms. When compare CBRG with LeastSquares, and Stochastic, the imputation time of CBRG is better than LeastSquares, and Stochastic in graduate admissions, diabetes, BNG (10000), BNG (15000), BNG (20000), Poker (10000) and Poker (50000) datasets, and worse when implemented on the rest of the datasets. In all missingness mechanisms, the imputation time of CBRG is better than MICE in all datasets. When compare CBRG with EMI, in MAR, the imputation time of CBRG is better than EMI in all datasets except in the graduate admissions, diabetes, profit estimation of companies and California datasets. When compare CBRG with Norm, in MAR, the imputation time of CBRG is better than EMI in all datasets except in the graduate admissions, diabetes, profit estimation of companies and California datasets. When compare CBRG with Fast KNN, in MAR, the imputation time of CBRG is better than Fast KNN in all datasets except in profit estimation of companies, red & white wine, California and diamonds datasets.

Choosing the candidate feature to be imputed by the CBRG depends on the feature selection of the GR. Calculating the GR requires calculating the entropy which is computationally expensive. Nevertheless, CBRG shows a good imputation time with the generated samples, from the Poker Hand and BNG_heart_statlog datasets. In addition, when manipulating too small size datasets, CBRG presents a good imputation time. The proposed algorithm consumes long imputation time when manipulating large size datasets. The MICE is not efficient in imputation time. Norm offers the best imputation time with all datasets. Fast KNN and EMI offer a good imputation time with small datasets but consume long imputation time when dealing with large datasets.

D. LIMITATIONS OF THE PROPOSED METHOD

The proposed method deals only with numerical features not nominal features. Imputation of large datasets leads the proposed method to consume more time than other stated methods. The proposed method assumes that the distribution of independent and dependent features is Gaussian distribution so it is affected with the distribution of the features.
### VI. CONCLUSION

It is essential to manipulate missing data, as it happens in almost all real world data. Manipulating incomplete instances is very significant for observational analyses with several predictors. In this work, the authors studied a set of previously published approaches used for manipulating missing data, reviewed their implementation on different datasets with different proportions of missing values were generated from the three missingness mechanisms. In addition, proposing a new algorithm CBRG works in cumulative order to impute all missing values. The candidate attribute is selected depending on the GR feature selection. The proposed algorithm shows good accuracy performance when compared with Stochastic, Fast KNN, Norm, MICE, LeastSquares, and EMI. The CBRG, LeastSquares, and MICE present the best performance among other mentioned methods. The proposed algorithm shows an acceptable running time and is considered a fast method but not the fastest because of the time consumed in calculating the GR. The results also revealed that the proposed algorithm works well with any missingness mechanism and with any missing data percentage. When the data features are highly correlated, the CBRG shows high imputation accuracy with low error as shown when implemented on the profit estimation of companies dataset.

In future research, it is recommended to implement the proposed imputation algorithm on additional datasets; additional units of standard error (like T-value and P-value) will be considered when choosing the candidate feature. The best future trend is to take the help of algorithms that deal with optimization problems with mixed features such as the GSA-GA algorithm [48]. In addition, the proposed algorithm is promising to be applied in imputing incomplete medical datasets, such as DNA microarray data, cardiovascular disease data, pulmonary embolism data, food composition data, and other medical data.

#### APPENDIX A

Tables 3 to 16 here.
W.-C. Lin and C.-F. Tsai, “Missing value imputation: A review and analysis of the literature (2006–2017),” Pattern Recognit., vol. 41, no. 12, pp. 3692–3705, Dec. 2008.

G. Varoquaux, L. Buitinck, G. Louppe, O. Grisel, F. Pedregosa, and A. Mueller, “Scikit-learn,” J. Mach. Learn. Res., vol. 12, no. 1, pp. 2825–2830, 2011.

Y. Li and L. E. Parker, “Nearest neighbor imputation using spatial–temporal correlations in wireless sensor networks,” Inf. Fusion, vol. 15, pp. 64–79, Jan. 2014.

L. Beretta and A. Santaniello, “Nearest neighbor imputation algorithms: A critical evaluation,” BMC Med. Informat. Decis. Making, vol. 16, no. 83, pp. 198–208, Jul. 2016.

A. Farhangfar, L. A. Kurgan, and W. Pedrycz, “A novel framework for imputation of missing values in databases,” IEEE Trans. Syst., Man, Cybern. A, Syst. Hum., vol. 37, no. 5, pp. 692–709, Sep. 2007.

R. Razavi-Far, B. Cheng, M. Saif, and A. Ahmadi, “Similarity-learning information-fusion schemes for missing data imputation,” Knowl-Based Syst., vol. 187, Jan. 2020, Art. no. 104805.

M. G. Rahman and M. Z. Islam, “Missing value imputation using a fuzzy clustering-based EM approach,” Knowl. Inf. Syst., vol. 46, no. 2, pp. 389–422, Feb. 2016.

C. B. Do and S. Batzoglou, “What is the expectation maximization algorithm?” Nature Biotechnol., vol. 26, no. 8, pp. 897–899, Aug. 2008.

Z. Ma and G. Chen, “Bayesian methods for dealing with missing data problems,” J. Korean Stat. Soc., vol. 47, no. 3, pp. 297–313, Sep. 2018.

H. Jiang, “Defect features recognition in 3D industrial CT images,” Informatika, vol. 42, no. 3, pp. 472–482, Sep. 2018.

A. C. Acocell, “Working with missing values,” J. Marriage Family, vol. 67, no. 4, pp. 1012–1028, Nov. 2005.

N. J. Horton and S. R. Lipsitz, “Multiple imputation in practice: Comparison of software packages for regression models with missing variables,” Amer. Statistician, vol. 55, no. 3, pp. 244–254, Aug. 2001.

M. Fichman and J. N. Cummings, “Multiple imputation for missing data: Making the most of what you know,” Organizational Res. Methods, vol. 6, no. 3, pp. 282–308, Jul. 2003.

J. W. Graham, “Missing data analysis: Making it work in the real world,” Annu. Rev. Psychol., vol. 60, no. 1, pp. 549–576, Jan. 2009.

Q. Chen, M. C. Paik, M. Kim, and C. Wang, “Using link-preserving imputation for logistic partially linear models with missing covariates,” Comput. Statist. Data Anal., vol. 101, pp. 174–185, Sep. 2016.

K. J. Lee and J. B. Carlin, “Multiple imputation for missing data: Fully conditional specification versus multivariate normal imputation,” Amer. J. Epidemiology, vol. 171, no. 5, pp. 624–632, Mar. 2010.

Z. Sahri, R. Yusof, and J. Watada, “FINNIM: Iterative imputation of missing values in dissolved gas analysis dataset,” IEEE Trans. Ind. Informat., vol. 10, no. 4, pp. 2093–2102, Nov. 2014.

L. F. Burgette and J. P. Reiter, “Multiple imputation for missing data via sequential regression trees,” Amer. J. Epidemiology, vol. 172, no. 9, pp. 1070–1076, Nov. 2010.

S. M. Mostafa et al.: CBRG: A Novel Algorithm for Handling Missing Data

S. M. Mostafa received the bachelor’s and M.Sc. degrees in computer science from the Computer Science-Mathematics Department, Faculty of Science, South Valley University, in 2004 and 2010, respectively, and the Ph.D. degree in computer science from the Advanced Information Technology Department, Graduate School of Information Technology, Kyushu University, Japan, in 2017. He is currently a Fellow with the Academy of Scientific Research and Technology (ASRT), Egypt. His research interests include machine learning and CPU scheduling.

ABDELRAHMAN S. ELADIMY received the bachelor’s degree in computer science from the Computer Science-Mathematics Department, Faculty of Science, South Valley University, in 2016. He is currently a Teaching Assistant. His research interests include machine learning and deep learning.
SAFWAT HAMAD received the bachelor’s degree, in 2000, the M.Sc. degree in modeling, simulation, and visualization, and the joint Ph.D. degree in high-performance computing from the Computer Science and Engineering Department, University of Connecticut, USA, and the Faculty of Computer and Information Sciences (FCIS), Ain Shams University (ASU), Cairo, Egypt, in 2008. He was a teaching assistant for several undergraduate courses. He has been the Chair of the Scientific Computing Department, FCIS, ASU, since 2017. He is currently an Associate Professor. His research interests include image and video processing, computational biology, machine learning, encryption, and security.

HIROFUMI AMANO (Member, IEEE) received the B.E. degree in electronics and the M.E. and D.Eng. degrees in computer science and communication engineering from Kyushu University, Fukuoka, Japan, in 1986, 1988, and 1991, respectively. From 1991 to 1994, he was a Research Associate with the Department of Computer Science and Communication Engineering, Kyushu University. In 1994, he was promoted to an Associate Professor with the Computer Center, Kyushu University. Since 2007, he has been an Associate Professor with the Research Institute for Information Technology, Kyushu University. He is currently working with the Graduate School of Information Science and Electrical Engineering, Kyushu University. His research interests include parallel processing, distributed processing, grid computing, and cloud computing. He is also a member of the Information Processing Society of Japan, and the Institute of Electronics, Information and Communication Engineers, Japan.