STATISTICS, OPTIMIZATION AND INFORMATION COMPUTING
*Stat., Optim. Inf. Comput.*, Vol. **14**, December 2025, pp 3526–3545.
Published online in International Academic Press (www.IAPress.org)

# On a new stacked ensemble framework for imputing missing data in the presence of outliers

Mahmoud A. Abdel-Fattah [*], Mai A. Mohsen, Amany M. Mousa

*Department of Applied Statistics and Econometrics, Faculty of Graduate Studies for Statistical Research, 5 Ahmed Zewail St., Cairo University, Giza 12613, Egypt*

**Abstract** Missing value imputation (MVI) presents a real challenge which becomes more complicated in the presence of outliers. Although ensemble techniques such as bagging and boosting have been employed for MVI and have shown promising results, stacking has not been investigated in this area, despite its efficiency in prediction tasks. To address this gap, two robust stacking frameworks are proposed for imputing missing data in the presence of outliers, namely RKSF-IM and RESF-IM. These proposed frameworks begin by adding an outlier indicator. Then they employ two different stacking configurations, where MissForest, IRMI, and EM are the base learners, and their predicted values are used as inputs in ridge regression, which acts as a meta learner in the second layer. The RMSE, MAE, and Wasserstein distance metrics of the proposed frameworks are evaluated against those of the mean, median, XGBoost, EM, IRMI, KNN, MissForest, and SVM imputation methods using a simulation study and two real data applications. The simulation study considers different scenarios for missing rates and outliers. The study also investigates the impact of adding an outlier indicator on the performance of the different imputation methods. The proposed stacking configurations show better performance, under the simulation settings, than the competing methods in most scenarios. In addition, many existing imputation methods are further improved by including an outlier indicator variable.

**Keywords** Missing value imputation, Ensemble, Stacking, MissForest, IRMI, EM, Ridge

**MSC 2020 subject classifications** 62D10, 68T05

**DOI:** 10.19139/soic-2310-5070-2894

## 1. Introduction

Missing data is a complicated challenge that arises from different sources, such as non-response in surveys, machine error, incomplete data records, and data entry mistakes [38]. Such missingness typically makes data analysis more difficult, since it results in biased findings and the loss of valuable information [29]. Furthermore, traditional statistical methods are designed for complete datasets; consequently, the presence of missing data complicates the analysis [26].

Machine learning (ML)-based imputation algorithms have received significant attention in recent years because of their superior imputation performance. Although many studies have demonstrated the effectiveness of these ML techniques in MVI, the problem of imputing missing values using ML techniques in the presence of outliers has seldom been explored. In general, the presence of outliers in datasets during data imputation can distort results [39].

Bagging and boosting as ensembling methods have been extensively used for MVI, while stacking remains unexplored in this domain, despite its superiority in prediction and classification tasks [47]. To address this gap, we introduce a robust stacking-based framework for imputing missing data specifically designed to handle datasets

[*] Correspondence to: Mahmoud A. Abdel-Fattah (Email: m.abdelfattah@cu.edu.eg). Department of Applied Statistics and Econometrics, Faculty of Graduate Studies for Statistical Research, 5 Ahmed Zewail St., Cairo University, Giza 12613, Egypt.

ISSN 2310-5070 (online) ISSN 2311-004X (print)

with outlier existence. Two stacking configurations, namely, robust K-folds stacking-based framework for imputing missing data (RKSF-IM) and robust enhanced stacking framework for imputing missing data (RESF-IM), are introduced. Results are evaluated against widely used imputation techniques. The evaluation is done in terms of the RMSE, MAE, and Wasserstein distance using a simulation study and two real data applications, considering different missing rates and outlier levels.

Additionally, this paper investigates the impact of adding an outlier indicator on the performance of widely used imputation methods. This indicator flags whether an observation is detected as an outlier. It can assess the impact of outlier awareness on imputation accuracy. This study also provides an evaluation of state-of-the-art imputation techniques in the presence of outliers.

The remainder of this paper is as follows. Section 2 presents an overview of ensemble methods. Section 3 introduces the proposed frameworks, while Section 4 describes the data sets, the simulation study settings and discusses the results. Section 5 provides the conclusion, limitations, and future work.

## 2. Ensemble learning

Traditional ML techniques usually train a single learner, while ensemble methods train multiple learners as base models and combine them to create a stronger overall model. These base learners can be any ML or statistical technique [51]. Combining more than one technique overcomes the limitations of a single individual prediction model and increases its forecasting accuracy [7].

[51, 19] demonstrated that combining multiple models gives more accurate results. It transforms weak learners into strong ones by training diverse base learners, as ensemble techniques incorporate different algorithms, model architectures, feature subsets, and bootstraps. Inspired by this encouraging performance, MVI has used ensemble techniques, particularly boosting and bagging, to enhance performance. Many studies have demonstrated that ensemble diverse models increase the imputation accuracy [17, 43].

Ensemble learning strategies can be classified into homogeneous and heterogeneous ensembles. Base learners of the same model type that have been trained using different parameters, random seeds, or subsets of the data (bootstraps) are included in homogeneous ensembles. In contrast, the base learners of heterogeneous ensembles are of different techniques, but they use the same training data [28]. Combining various models, whether they are of the same or different types, usually improves predictive accuracy and robustness. However, it needs careful architecture of the ensemble and a wise determination of the voting method.

Another classification of ensemble techniques is bagging, boosting, and stacking techniques. Bagging usually involves two main processes: bootstrapping, where the data is divided into different subsamples taken with replacement by resampling techniques, each bootstrap is used as training data, and learners are run in parallel on these different bootstraps. By creating more training data from the original data, this lowers the variance [15]. The second process is aggregation, where predictions are aggregated by averaging or voting. Bagging has been intensively used in MVI [11, 33, 42, 24]. One of the most well-known MVI techniques that implies bagging is MissForest [37].

For boosting, it trains weak learners sequentially, where each learner corrects the errors of the previous one. It turns those weak learners into strong ones. Generally, boosting is a numerical optimization technique that is used to minimize the loss function [16]. Boosting has also been used as an MVI technique [48, 49, 27]. The results of these studies were very promising.

For stacking, all the training data is used to train the base models (level 1 learners), and the predictions of these base learners are fed to a second-level model called the meta learner. So, the meta learner is trained on the output of the base models [35]. The meta model is often chosen to be a linear model, such as ridge and lasso regressions, as they can provide a smooth interpretation of the predictions of base models [50]. To avoid overfitting and information leaking, an advanced version of stacking applies cross-validation techniques to generate the meta learner training set. This enhanced approach utilizes leave-one-out cross-validation or K-folds cross-validation procedures. When performing K-folds cross validation, base learners are trained on subsets of the data, usually one-kth of the original dataset is left out, and predictions are made on the hold-out fold to form the meta learner features. This process is repeated until each observation is utilized exactly once for validation [14]. Algorithm 1 illustrates how stacking with K-folds cross validation works.

**Algorithm 1** Stacking ensemble with K-folds cross-validation

```
Require: Training dataset $D$, base learners $\{B_1, B_2, \ldots, B_L\}$
Ensure: Stacked ensemble (trained meta learner)
Initialize: $M \leftarrow \emptyset$ (meta-features matrix), $k \leftarrow$ number of folds
Split $D$ randomly into $k$ equal subsets: $D = \{D_1, D_2, \ldots, D_k\}$
for each fold $j$ from 1 to $k$ do
    for each base learner $h_i$ from 1 to $L$ do
        Train $h_i$ on $D \setminus D_j$ (all folds except $j$th fold)
        Predict: $\hat{y}_j^{(i)} \leftarrow h_i(D_j)$ (predictions of $j$th fold)
        Store $\hat{y}_j^{(i)}$
    end for
end for
$M$ is now $n \times (L+1)$ matrix where $n = |D|$ and $L =$ number of base learners
Train the meta learner on $M$ using the original target labels $y$
```

Stacking has rarely been used to deal with missing values for classification problems [34, 8] and to directly incorporate missing data into the analysis model [12]. However, it has not yet been explored as an imputation technique by itself that returns a complete dataset for use in any analysis.

## 3. Proposed hybrid frameworks

Two hybrid robust frameworks are proposed for imputing missing data in the presence of outliers. Both of them share the same first stage, which involves creating an outlier indicator variable (Algorithm 2) to indicate whether an observation is considered an outlier or not, thereby benefiting from this information during the imputation. The inclusion of the outlier indicator is examined for individual MVI techniques. The two proposed methods differ in the second stage, as the first method is adopting the traditional stacking for handling missingness problem as a prediction problem, especially in the presence of outliers. This framework is referred to as RKSF-IM, while the second proposed method examines a new stacking configuration, as we will discuss. This framework will be referred to as RESF-IM.

### *3.1. Robust K-folds stacking framework for imputing missing data (RKSF-IM)*

The proposed framework adapts the stacking ensemble for imputing missing values in the presence of outliers as follows.

**First stage: adding an outlier indicator variable**, where outliers are detected as follows:

1. Missing data is filled by a robust placeholder value that is the median.
2. Outliers are detected by density-based spatial clustering of applications with noise (DBSCAN); any other outlier detection technique can be used.
3. A new variable is added, which determines, according to the DBSCAN, whether an observation is an outlier or not.

**Second stage: adapting the stacking framework for MVI**, for each variable $y_j$ with missing values:

1. Missing data is again set to be missing for $y_j$ which becomes a target variable.
2. Data is separated into complete data $D_j^{\text{obs}}$ (rows where $y_j$ is observed) vs missing data $D_j^{\text{miss}}$ (rows where $y_j$ is missing).

**Algorithm 2** Adding an outlier indicator variable (first stage)

```
Require: Dataset matrix $D$ with missing values
Ensure: $D$ with added outlier indicator variable
for each variable $X_j$ in $D$ do
    Replace missing values with the median as a robust placeholder
end for
Apply DBSCAN on $D$ (with temporarily imputed values)
for each observation $i$ in $D$ do
    if observation $i$ is detected as an outlier by DBSCAN then
        Outlier indicator $O_i \leftarrow 1$
    else
        Outlier indicator $O_i \leftarrow 0$
    end if
end for
Append outlier indicator as a new variable to $D$
```

3. Complete data $D_j^{\text{obs}}$ becomes the training data which is split into $k$ folds.
4. For each base learner, data is trained on $k-1$ folds and one fold for validation, rotate until each fold is used as a validation fold.
5. Record the predictions made by each validation fold and combine them in one column (so one column for each base learner).
6. The predictions $\hat{y}_{j/\text{base}_i}^{\text{obs}}$ are collected from the $L$ base learners where $L$ is the number of base learners in level 1. These predictions $\hat{y}_{j/\text{base}_i}^{\text{obs}}$ enter the meta learner each as a variable with the original values of $y_j$ as the target variable to make the final $\hat{y}_{j/\text{meta}}^{\text{obs}}$ prediction.
7. Apply each fully trained base model to the incomplete data to make predictions for $\hat{y}_{j/\text{base}_i}^{\text{miss}}$.
8. Apply the trained meta learner by feeding it with these base learner predictions $\hat{y}_{j/\text{base}_i}^{\text{miss}}$ to make the final predictions $\hat{y}_{j/\text{meta}}^{\text{miss}}$ for the missing values in $y_j$.

**Algorithm 3** Proposed RKSF-IM

```
Require: Dataset matrix $D$ with missing values, base learners $\{B_1, B_2, \ldots, B_L\}$, number of folds $K$, meta learner $M$
Ensure: Fully imputed dataset $D^*$
Add an outlier indicator variable as described by Algorithm 2
for each variable $X_j$ in $D$ with missing values do
    Set the placeholder-imputed values back to missing (becomes target $y$)
    Partition dataset:
        $D_{\text{obs}} \leftarrow$ rows where $X_j$ is observed
        $D_{\text{miss}} \leftarrow$ rows where $X_j$ is missing
    Train stacking ensemble $K$-folds cross validation (Algorithm 1) on $D_{\text{obs}}$
    Predict missing values of $X_j$ in $D_{\text{miss}}$ : $\hat{X}_j^{\text{miss}}$
    Update $D$ by replacing missing values in $X_j$ with $\hat{X}_j^{\text{miss}}$
end for
```

### *3.2. Robust enhanced stacking framework for imputing missing data (RESF-IM)*

This stacking framework has two stages. It shares the same first stage (Algorithm 2) with RKSF-IM, but differs in the second stage. For every variable $y_j$with m missing values:

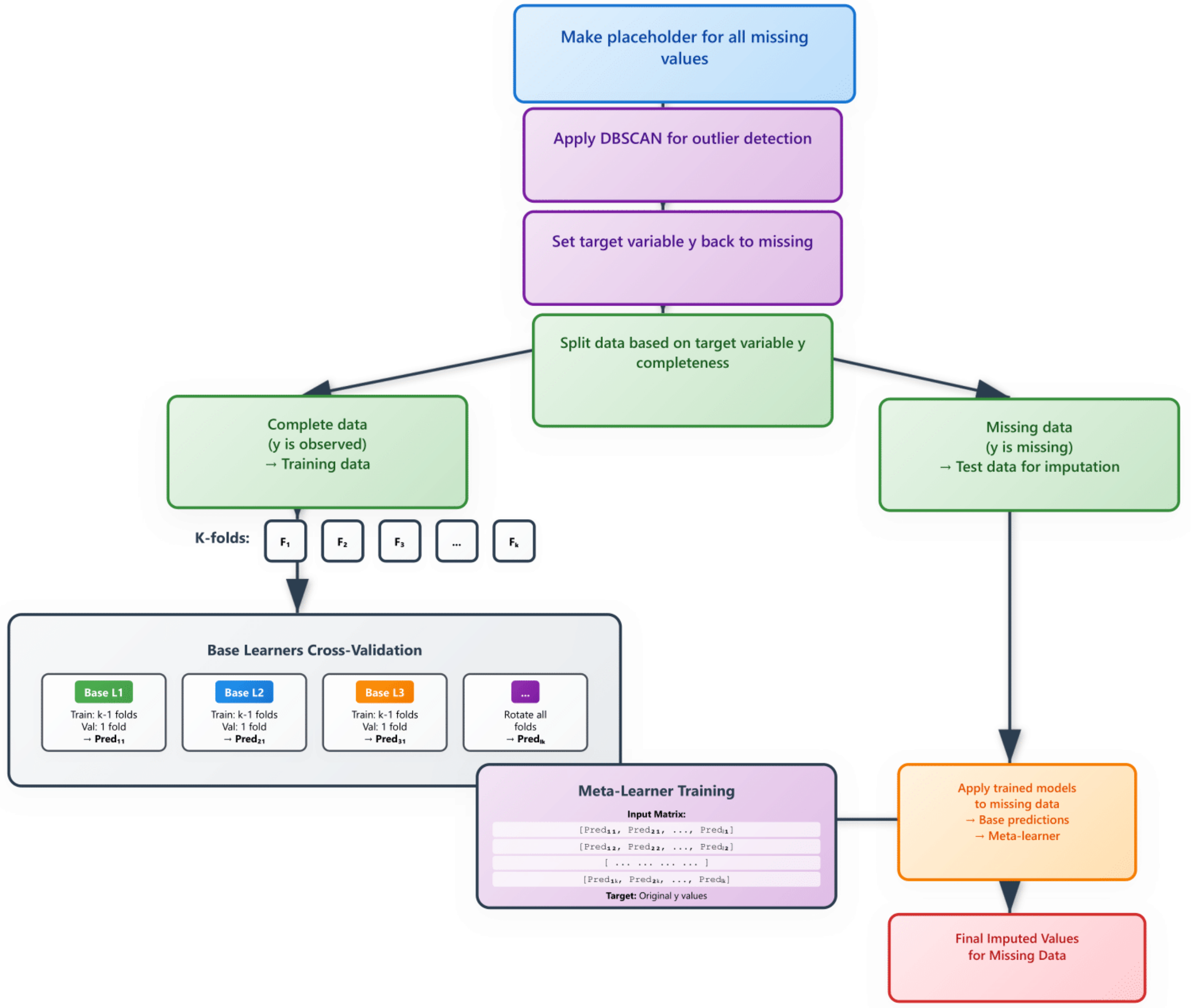
Make placeholder for all missing values
Apply DBSCAN for outlier detection
Set target variable y back to missing
Split data based on target variable y completeness
Complete data (y is observed) → Training data
Missing data (y is missing) → Test data for imputation
K-folds:
$F_1$
$F_2$
$F_3$
...
$F_k$
Base Learners Cross-Validation
Base L1
Train: k-1 folds Val: 1 fold → $\mathrm{Pred}_{11}$
Base L2
Train: k-1 folds Val: 1 fold → $\mathrm{Pred}_{21}$
Base L3
Train: k-1 folds Val: 1 fold → $\mathrm{Pred}_{31}$
...
Rotate all folds → $\mathrm{Pred}_{lk}$
Meta-Learner Training
Input Matrix:
$[\mathrm{Pred}_{11}, \mathrm{Pred}_{21}, \ldots, \mathrm{Pred}_{l1}]$
$[\mathrm{Pred}_{12}, \mathrm{Pred}_{22}, \ldots, \mathrm{Pred}_{l2}]$
[ ... ... ... ... ]
$[\mathrm{Pred}_{1k}, \mathrm{Pred}_{2k}, \ldots, \mathrm{Pred}_{lk}]$
Target: Original y values
Apply trained models to missing data → Base predictions → Meta-learner
Final Imputed Values for Missing Data

Figure 1. The proposed RKSF-IM.

1. Missing data is again set to be missing for $y_j$ which becomes a target variable, data is separated to complete data $D_j^{\text{obs}}$ (rows where $y_j$ is observed) versus missing data $D_j^{\text{miss}}$ (rows where $y_j$ is missing).
2. Missing values in $Y_j^{\text{miss}}$ are imputed using $L$ (here $l = 3$) base learners.
3. The meta learner training data is then built by four parts stacked as rows over each other:
   - part A: $D_j^{\text{obs}}$ (rows where $y_j$ is observed).
   - Part B: $\hat{D}_{j/\text{base}_1}^{\text{miss}}$ rows where the missing $y_j$ is filled by its predicted value from the base learner 1 (not only $\hat{y}_j$ values of as the previous framework but the whole row).
   - Parts C and D are as part B but using the other two base learners.
4. The meta learner is trained on the training data, and the missing values are finally predicted $\hat{y}_{j/\text{meta}}^{\text{miss}}$. This method iterates over each variable with missing values in a randomized order.

The frameworks employ IRMI, EM, and MissForest as base learners. IRMI was introduced by Templ et al. [40] as a stepwise regression imputation that can use robust methods for different types of data. Robust methods are needed when the multivariate normal (MVN) assumption is violated by the existence of outliers in the data.

**Algorithm 4** Robust enhanced stacking framework for imputing missing data (RESF-IM)

```
Require: Dataset matrix $D$ with missing values, base learners $\{B_1, B_2, \ldots, B_L\}$, number of folds $K$, meta learner $M$
Ensure: $\hat{D}$ (Fully imputed dataset)
Stage 1: Add outlier indicator variable (same as Algorithm 2)
Stage 2: Enhanced stacking-based imputation
for each variable $y_j$ in $D$ with missing values do
    Set the placeholder imputed values back missing (becomes target $x$)
    Partition dataset:
        $D_j^{\text{obs}} \leftarrow$ rows where $y_j$ is observed
        $D_j^{\text{miss}} \leftarrow$ rows where $y_j$ is missing
    for each base learner $B_i \in \{B_1, B_2, \ldots, B_L\}$ do
        Train $B_i$ on $D_j^{\text{obs}}$
        Apply trained $B_i$ on $D_j^{\text{miss}}$
        Return $\hat{D}_{j/\text{base}_i} \leftarrow$ full dataset imputed by base learner $i$
    end for
    Training data $\leftarrow \{\}$
    part A $\leftarrow D_j^{\text{obs}}$ (rows where $y_j$ is observed)
    part B $\leftarrow \hat{D}_{j/\text{base}_1}$ (full rows with $y_j$ filled by $B_1$ prediction)
    Training data $\leftarrow$ Stack $\begin{pmatrix} \text{Part A} \\ \text{Part B} \end{pmatrix}$ (vertically)
    Train meta learner $M$ on (Training data)
    Predict: $\hat{y}_j$ missing values $\leftarrow M(D_j^{\text{miss}})$
    Update $D$ by replacing missing values in $y_j$ with $\hat{y}_j$
end for
```

IRMI uses the missing value as a target value and the remaining variables as regressors. It iterates between model estimation and MVI until convergence. Although all the instances are used for imputing the missing values, the imputed values are less influenced by outliers because of using robust statistical techniques.

The EM approach is a well-established statistical technique for MVI. It is an iterative maximum likelihood estimation that repeats expectation (E-step) and maximization (M-step) for incomplete data [13].

MissForest is a random forest (RF)-based technique that treats the missing values problem as a prediction problem. It inherent RF efficiency in handling complex interactions and nonlinear relationships without requiring explicit model specification. MissForest was first introduced by Stekhoven and Bühlmann [37].

To ensure robustness and validate the findings, an alternative configuration is implemented using the same base learners, except EM is substituted by eXtreme gradient boosting (XGBoost).

XGBoost is an effective gradient boosting framework that combines multiple weak learners; these weak learners correct the errors of the previous ones. It can handle nonlinear relationships, mixed data, and high-dimensional datasets [44].

Stacking faces multicollinearity as the base learners' predictions are multicollinear. Ridge regression is a shrinkage method that can handle the multicollinearity problem in regression analysis [22]. Ridge-type estimation has been extensively studied in various regression research [1, 2, 3, 4]. Hence, ridge regression is trained as the meta-learner to tackle the multicollinearity problem, and the ridge constant is computed using cross-validation.

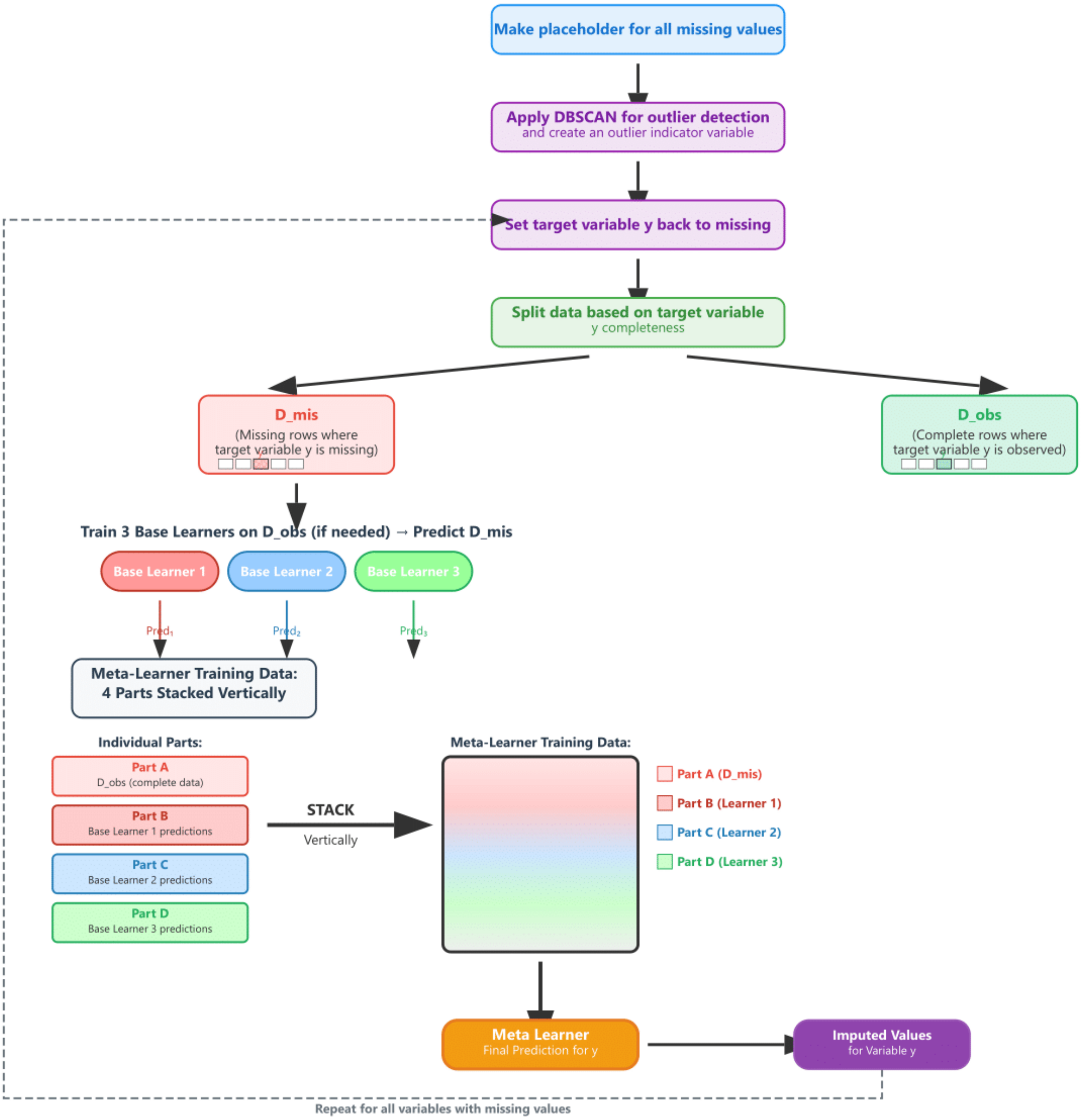
Make placeholder for all missing values
Apply DBSCAN for outlier detection
and create an outlier indicator variable
Set target variable y back to missing
Split data based on target variable
y completeness
D_mis
(Missing rows where
target variable y is missing)
D_obs
(Complete rows where
target variable y is observed)
Train 3 Base Learners on D_obs (if needed) → Predict D_mis
Base Learner 1
Base Learner 2
Base Learner 3
$\text{Pred}_1$
$\text{Pred}_2$
$\text{Pred}_3$
Meta-Learner Training Data:
4 Parts Stacked Vertically
Individual Parts:
Part A
D_obs (complete data)
Part B
Base Learner 1 predictions
Part C
Base Learner 2 predictions
Part D
Base Learner 3 predictions
STACK
Vertically
Meta-Learner Training Data:
Part A (D_mis)
Part B (Learner 1)
Part C (Learner 2)
Part D (Learner 3)
Meta Learner
Final Prediction for y
Imputed Values
for Variable y
Repeat for all variables with missing values

Figure 2. The proposed RESF-IM.

## 4. Simulation study and applications

The proposed methods are evaluated using a simulation study and two real data applications. For comparative analysis, results are compared to Missforest, IRMI, EM, XGBoost, mean imputation, median imputation, K-nearest neigbour (KNN), and support vector machine (SVM).

Mean imputation, where each missing value in variable $X_j$ is filled by the mean of the variable $X_j$, median imputation is the same, but the median is used to fill the missing holes. KNN is a supervised learning method where the values estimated from the k nearest observed data are used to impute missing values. The nearest neighbors can be determined using a distance function, which is often the Euclidean distance [21]. In general, KNN is one of the most popular MVI methods based on ML because of its simplicity, power, and competitive outcomes. SVM is a ML method that finds an optimal hyperplane by maximizing the margin between data points. Variables with missing values are treated as a target variable to be predicted. Through kernel functions, SVMs capture complex nonlinear relationships and effectively handle high-dimensional data [5].

Each imputation method is trained once with the outlier indicator included and once without it, except for IRMI, which is already robust to outliers. The following three evaluation metrics are used [10, 30].

1. **Root mean squared error (RMSE)**:

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

2. **Mean absolute error (MAE)**:

$$\text{MAE} = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i|. \tag{2}$$

3. **Wasserstein $l_2$ distance** which assesses how similar the distributions of imputed and original data are for each variable [41]. It is calculated as in [52].

### 4.1. Simulation settings

Data is generated from the contaminated MVN, which represents a simple heavy-tailed generalization of the MVN using different outlier rates (5%, 10%, 20%) as follows:

$$\alpha\%\mathcal{N}(\mu, \Sigma) + (100 - \alpha)\%\mathcal{N}(\mu, k\Sigma), \tag{3}$$

where $\alpha$ is the proportion of normal observations, $\mu$ is the mean vector, $\Sigma$ is the covariance matrix, and $k$ is the contamination factor. The sample was drawn from two distinct MVN distributions; a MVN $(\mu_1, \Sigma)$ distribution with probability $\alpha$ and from a MVN $(\mu_1, k\Sigma)$ distribution with probability $1 - \alpha$, where $k > 1$ is a constant that represents the size of the contaminated component $k$ [45] which is chosen as 20, 50 and 100. So, outliers are generated with a covariance matrix that is (twenty, fifty, hundred) times larger, introducing greater variability and making them distinct from normal observations. $\mu$ and $\Sigma$ were chosen from a real data set [32] that is Boston housing dataset [20] where only 5 variables were selected:

$$\boldsymbol{\mu} = (15.0, 11.5, 32.1, 36.4, 14.3), \tag{4}$$

$$\boldsymbol{\Sigma} = \begin{bmatrix} 12.71 & 4.79 & 9.46 & 10.92 & 5.91 \\ 4.79 & 8.44 & 5.38 & 6.81 & 3.69 \\ 9.46 & 5.38 & 20.72 & 23.15 & 8.64 \\ 10.92 & 6.81 & 23.15 & 31.15 & 10.93 \\ 5.91 & 3.69 & 8.64 & 10.93 & 7.06 \end{bmatrix}. \tag{5}$$

Different missing rates are artificially introduced under missing completely at random (MCAR) at 5%, 10%, 20% and 30% missing rates. Under MCAR a missing value in a specific attribute doesn't depend on either the known values or the missing data value itself, that is if

$$P(R|Y_{\text{obs}}, Y_{\text{mis}}, \Phi) = P(R|\Phi), \tag{6}$$

where $\Phi$ refers to the parameters of the missing data mechanism, $Y_{\text{obs}}$ are the observed values of $Y$, and $Y_{\text{mis}}$ are the missing values of $Y$, $R$ is a binary variable that indicates whether a value is observed or missing for each data point [26]. The simulation study considers two different sample sizes (200 and 500 observations) and five variables. The dataset is randomly shuffled to ensure no systematic ordering of normal and outlier points, simulating real datasets. For each scenario, 500-run were generated as in [6, 9, 36], with a total of 72 scenarios. After artificially creating missing values, these missingness are imputed using the median as a placeholder to apply DBSCAN and detect the outliers and to standardize the data, as many imputation techniques need standardized data. After that, missing values are reset missing. For DBSCAN parameters, minpts are determined using the rule dimension + 1, while eps is determined via a pilot study using the k-distance graph method. The k-nearest neighbor distances for all points are ordered in descending order and identified the elbow point in the sorted distance plot to define the eps value. We will present results for $k = 20$. Results for $k = 50$ and $k = 100$ support the results and will be made available upon reasonable request from the corresponding author.

### 4.2. Simulation results

Table 1. Simulated RMSE for different imputation methods across different missing rates with 5%, 10%, and 20% contamination for n = 200 and k = 20.

| | 5% contamination | | | | 10% contamination | | | | 20% contamination | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Missing rate | 5% | 10% | 20% | 30% | 5% | 10% | 20% | 30% | 5% | 10% | 20% | 30% |
| Mean | 0.1696 | 0.2902 | 0.4357 | 0.5375 | 0.1635 | 0.2872 | 0.4367 | 0.5383 | 0.1581 | 0.2901 | 0.4398 | 0.5478 |
| Median | 0.1696 | 0.2903 | 0.4352 | 0.5373 | 0.1636 | 0.2865 | 0.4362 | 0.5375 | 0.1578 | 0.2889 | 0.4392 | 0.5467 |
| KNN (OI) | 0.1276 | 0.2284 | 0.3581 | 0.4598 | 0.1275 | 0.2314 | 0.3799 | 0.4821 | 0.1276 | 0.2262 | 0.3794 | 0.4765 |
| KNN (WOI) | 0.1275 | 0.2293 | 0.3591 | 0.4609 | 0.1246 | 0.2314 | 0.3709 | 0.4724 | 0.1233 | 0.2284 | 0.373 | 0.4819 |
| IRMI | 0.1445 | 0.2277 | 0.3447 | 0.4461 | 0.136 | 0.2186 | 0.3429 | 0.5363 | 0.1299 | 0.2145 | 0.3434 | 0.5491 |
| MissForest (OI) | 🟧 0.1191 | 🟧 0.2078 | 🟧 0.3277 | 🟧 0.4217 | 🟨 0.1146 | 🟧 0.2067 | 🟧 0.3347 | 🟧 0.4328 | 🟨 0.1144 | 🟧 0.2036 | 🟧 0.3364 | 🟨 0.415 |
| MissForest (WOI) | 🟧 0.1196 | 🟧 0.2047 | 🟧 0.3283 | 🟨 0.4215 | 🟨 0.1135 | 🟧 0.2066 | 🟧 0.3353 | 🟧 0.4324 | 🟨 0.1175 | 🟧 0.2051 | 🟧 0.3361 | 🟨 0.4186 |
| XGBoost (OI) | 0.1264 | 0.2169 | 0.3502 | 0.4549 | 0.1219 | 0.2187 | 0.3481 | 0.4547 | 0.1201 | 0.2192 | 0.3503 | 0.4536 |
| XGBoost (WOI) | 0.1264 | 0.2197 | 0.3531 | 0.4606 | 0.1218 | 0.2221 | 0.3545 | 0.4587 | 0.1199 | 0.2204 | 0.3577 | 0.4639 |
| SVM (OI) | 0.128 | 0.234 | 0.3776 | 0.4826 | 0.1274 | 0.2448 | 0.3986 | 0.5039 | 0.1273 | 0.248 | 0.4025 | 0.5193 |
| SVM (WOI) | 0.1289 | 0.2342 | 0.3776 | 0.4854 | 0.1272 | 0.2448 | 0.3995 | 0.505 | 0.1269 | 0.2481 | 0.4019 | 0.5193 |
| EM (OI) | 0.1664 | 0.2684 | 0.4266 | 0.5385 | 0.1708 | 0.2741 | 0.4221 | 0.5425 | 0.1617 | 0.2742 | 0.4284 | 0.5413 |
| EM (WOI) | 0.1618 | 0.2673 | 0.4194 | 0.5346 | 0.1625 | 0.2748 | 0.4181 | 0.5425 | 0.1666 | 0.2674 | 0.4221 | 0.5376 |
| RESF-IM(Xgb) | 🟩 0.1132 | 🟩 0.1947 | 🟩 0.3119 | 🟩 0.4049 | 🟩 0.1091 | 🟩 0.1924 | 🟩 0.3087 | 🟩 0.4121 | 🟩 0.1043 | 🟩 0.191 | 🟩 0.313 | 🟩 0.4135 |
| RESF-IM (EM) | 🟩 0.1112 | 🟩 0.193 | 🟩 0.3096 | 🟩 0.4001 | 🟩 0.1084 | 🟩 0.1911 | 🟩 0.3054 | 🟩 0.4083 | 🟩 0.104 | 🟩 0.1891 | 🟩 0.3113 | 🟩 0.4107 |
| RKSF-IM (EM) | 🟨 0.1184 | 🟨 0.2032 | 🟨 0.3259 | 🟨 0.4204 | 🟧 0.117 | 🟨 0.2035 | 🟨 0.3249 | 🟨 0.4231 | 🟨 0.1111 | 🟨 0.1993 | 🟨 0.3282 | 🟧 0.4304 |
| RKSF-IM (Xgb) | 🟨 0.1183 | 🟨 0.2044 | 🟨 0.3274 | 🟧 0.4244 | 🟧 0.1172 | 🟨 0.2045 | 🟨 0.3268 | 🟨 0.4253 | 🟨 0.1127 | 🟨 0.2022 | 🟨 0.3279 | 🟧 0.4345 |

*OI indicates with outlier indicator, WOI indicates without outlier indicator.

🟩 First and second ranks 🟨 Third and fourth ranks 🟧 Fifth and sixth ranks

- **Overall performance for n = 200 and k = 20:**
  Tables 1, 2, and 3 present the simulated RMSE, MAE, and Wasserstein distance for different imputation methods across different missing rates and contamination levels for n = 200 and k = 50. It can be concluded that the proposed RESF-IM consistently outperforms other algorithms for all performance evaluation metrics across different missing rates scenarios and for all contamination levels. Followed by RKSF-IM in most scenarios according to various evaluation metrics, with some variations as follows:
  - According to RMSE, the proposed RKSF-IM comes after RESF-IM for all missing rates, followed by MissForest for all contamination rates except low missing rates at 10% contamination and high missing rates for 20% contamination, where MissForest comes first.

Table 2. Simulated MAE for different imputation methods across different missing rates with 5%, 10%, and 20% contamination for n = 200 and k = 20.

| | 5% contamination | | | | 10% contamination | | | | 20% contamination | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Missing rate | 5% | 10% | 20% | 30% | 5% | 10% | 20% | 30% | 5% | 10% | 20% | 30% |
| Mean | 0.035 | 0.0689 | 0.1369 | 0.2044 | 0.331 | 0.651 | 0.1924 | 0.1924 | 0.0546 | 0.0636 | 0.1266 | 0.1894 |
| Median | 0.035 | 0.0689 | 0.1369 | 0.2043 | 0.332 | 0.649 | 0.192 | 0.192 | 0.0545 | 0.0633 | 0.126 | 0.1885 |
| KNN (OI) | 0.0258 | 0.0522 | 0.1063 | 0.166 | 0.0253 | 0.0501 | 0.1607 | 0.1607 | 0.0442 | 0.0484 | 0.1036 | 0.1583 |
| KNN (WOI) | 0.0258 | 0.0524 | 0.1064 | 0.1662 | 0.0225 | 0.0501 | 0.1594 | 0.1594 | 0.0426 | 0.0486 | 0.1022 | 0.1595 |
| IRMI | 0.0294 | 0.0554 | 0.1091 | 0.1647 | 0.0274 | 0.0509 | 0.1684 | 0.1684 | 0.0419 | 0.0492 | 0.1001 | 0.1688 |
| MissForest (OI) | 0.024 | 0.0473 | 0.0973 | 0.1522 | 0.0227 | 0.0451 | 0.0944 | 0.1471 | 0.0407 | 0.0435 | 0.0933 | 0.1397 |
| MissForest (WOI) | 0.0242 | 0.0469 | 0.0979 | 0.1531 | 0.0225 | 0.0451 | 0.1476 | 0.1476 | 0.0408 | 0.0439 | 0.0935 | 0.1408 |
| XGBoost (OI) | 0.0254 | 0.0501 | 0.1045 | 0.1648 | 0.0241 | 0.048 | 0.0992 | 0.1558 | 0.0416 | 0.0469 | 0.0981 | 0.1538 |
| XGBoost (WOI) | 0.0257 | 0.0506 | 0.1058 | 0.1683 | 0.0242 | 0.0487 | 0.1013 | 0.1591 | 0.0418 | 0.0471 | 0.1005 | 0.1583 |
| SVM (OI) | 0.0254 | 0.051 | 0.1063 | 0.1674 | 0.0249 | 0.0504 | 0.1617 | 0.1617 | 0.0467 | 0.0499 | 0.1046 | 0.1639 |
| SVM (WOI) | 0.0257 | 0.0512 | 0.1069 | 0.1696 | 0.0248 | 0.0505 | 0.1626 | 0.1626 | 0.0465 | 0.0499 | 0.1045 | 0.1644 |
| EM (OI) | 0.0344 | 0.0657 | 0.1377 | 0.2196 | 0.0348 | 0.0659 | 0.2091 | 0.2091 | 0.0492 | 0.0662 | 0.1382 | 0.2142 |
| EM (WOI) | 0.0331 | 0.0658 | 0.1382 | 0.2143 | 0.0332 | 0.0664 | 0.2122 | 0.2122 | 0.0486 | 0.0647 | 0.1372 | 0.2136 |
| RESF-IM (Xgb) | 0.0229 | 0.0453 | 0.0944 | 0.1484 | 0.0217 | 0.0423 | 0.0892 | 0.1415 | 0.0367 | 0.0409 | 0.0873 | 0.1382 |
| RESF-IM (EM) | 0.0225 | 0.0451 | 0.0938 | 0.1468 | 0.0216 | 0.0422 | 0.1403 | 0.1399 | 0.0366 | 0.0407 | 0.0865 | 0.137 |
| RKSF-IM (EM) | 0.024 | 0.0477 | 0.099 | 0.1542 | 0.0233 | 0.0454 | 0.1464 | 0.1464 | 0.0383 | 0.0434 | 0.0926 | 0.1459 |
| RKSF-IM (Xgb) | 0.024 | 0.0477 | 0.099 | 0.1542 | 0.0233 | 0.0455 | 0.1466 | 0.1466 | 0.0385 | 0.0438 | 0.0919 | 0.1463 |

*OI indicates with outlier indicator, WOI indicates without outlier indicator.

🟩 First and second ranks 🟨 Third and fourth ranks 🟧 Fifth and sixth ranks

Table 3. Simulated Wasserstein distance for different imputation methods across different missing rates with 5%, 10%, and 20% contamination for n = 200 and k = 20.

| | 5% contamination | | | | 10% contamination | | | | 20% contamination | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Missing rate | 5% | 10% | 20% | 30% | 5% | 10% | 20% | 30% | 5% | 10% | 20% | 30% |
| Mean | 0.0504 | 0.0716 | 0.1018 | 0.1238 | 0.0501 | 0.0719 | 0.1007 | 0.123 | 0.0546 | 0.0778 | 0.1096 | 0.1328 |
| Median | 0.0504 | 0.0716 | 0.1018 | 0.1238 | 0.0501 | 0.0718 | 0.1006 | 0.1229 | 0.0545 | 0.0776 | 0.1094 | 0.1325 |
| KNN (OI) | 0.0392 | 0.0574 | 0.084 | 0.1051 | 0.0403 | 0.0591 | 0.0883 | 0.1096 | 0.0442 | 0.0629 | 0.0944 | 0.1154 |
| KNN (WOI) | 0.0393 | 0.0576 | 0.0842 | 0.1053 | 0.0399 | 0.0589 | 0.0863 | 0.1078 | 0.0426 | 0.0631 | 0.0931 | 0.1164 |
| IRMI | 0.0398 | 0.055 | 0.0771 | 0.0944 | 0.0381 | 0.0536 | 0.0785 | 0.0987 | 0.0419 | 0.0575 | 0.0842 | 0.1055 |
| MissForest (OI) | 0.0361 | 0.0525 | 0.0776 | 0.0964 | 0.0362 | 0.0527 | 0.0776 | 0.0982 | 0.0407 | 0.0567 | 0.0844 | 0.1007 |
| MissForest (WOI) | 0.036 | 0.0523 | 0.0773 | 0.0959 | 0.0359 | 0.0526 | 0.0776 | 0.0974 | 0.0408 | 0.0569 | 0.0842 | 0.1011 |
| XGBoost (OI) | 0.0383 | 0.0559 | 0.0817 | 0.102 | 0.0384 | 0.0553 | 0.1295 | 0.1018 | 0.0416 | 0.0604 | 0.087 | 0.1076 |
| XGBoost (WOI) | 0.0382 | 0.0559 | 0.0822 | 0.1026 | 0.0382 | 0.0559 | 0.0804 | 0.1027 | 0.0418 | 0.0609 | 0.0879 | 0.109 |
| SVM (OI) | 0.0414 | 0.0606 | 0.0894 | 0.1113 | 0.0436 | 0.0641 | 0.0813 | 0.1158 | 0.0467 | 0.0687 | 0.1011 | 0.1258 |
| SVM (WOI) | 0.0413 | 0.0605 | 0.0893 | 0.1118 | 0.0434 | 0.0639 | 0.0926 | 0.1159 | 0.0465 | 0.0686 | 0.1008 | 0.1256 |
| EM (OI) | 0.0447 | 0.0632 | 0.0888 | 0.1077 | 0.0446 | 0.0635 | 0.0896 | 0.1098 | 0.0492 | 0.0687 | 0.0977 | 0.1177 |
| EM (WOI) | 0.0447 | 0.0629 | 0.0879 | 0.1059 | 0.0446 | 0.0631 | 0.0885 | 0.1089 | 0.0486 | 0.0686 | 0.0967 | 0.1171 |
| RESF-IM (Xgb) | 0.034 | 0.0497 | 0.0741 | 0.0939 | 0.0336 | 0.0493 | 0.0725 | 0.0938 | 0.0367 | 0.0533 | 0.0792 | 0.1002 |
| RESF-IM (EM) | 0.0337 | 0.0491 | 0.0731 | 0.0924 | 0.0334 | 0.049 | 0.0717 | 0.0925 | 0.0366 | 0.053 | 0.0784 | 0.0992 |
| RKSF-IM (EM) | 0.0354 | 0.0518 | 0.0767 | 0.0971 | 0.0353 | 0.0516 | 0.0753 | 0.0966 | 0.0383 | 0.0557 | 0.0823 | 0.1042 |
| RKSF-IM (Xgb) | 0.0358 | 0.0521 | 0.0772 | 0.0978 | 0.0356 | 0.0519 | 0.0759 | 0.0972 | 0.0385 | 0.0561 | 0.0825 | 0.105 |

*OI indicates with outlier indicator, WOI indicates without outlier indicator.

🟩 First and second ranks 🟨 Third and fourth ranks 🟧 Fifth and sixth ranks

- For MAE, RKSF-IM performs better than MissForest at low missing rates at 5% contamination. In comparison, this is reversed at 10% contamination, where RKSF-IM performs better than MissForest at higher missing rates.
- According to Wasserstein distance, RKSF-IM is almost the second-best method at all contamination levels, except for a few scenarios with high missing rates, where MissForest and IRMI come before it.

Table 4. Simulation study results for the impact of adding an outlier indicator variable for different imputation methods across different missing and contamination rates according to RMSE, MAE and Wasserstein distance for n = 200 and k = 20.

| | | 5% contamination | | | | 10% contamination | | | | 20% contamination | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Missing rate | 5% | 10% | 20% | 30% | 5% | 10% | 20% | 30% | 5% | 10% | 20% | 30% |
| RMSE | KNN (OI) | ✗ | ✓ | ✓ | ✓ | ✗ | ✓ | ✗ | ✗ | ✗ | ✓ | ✗ | ✓ |
| | MissForest (OI) | ✓ | ✗ | ✓ | ✗ | ✗ | ✗ | ✓ | ✓ | ✓ | ✓ | ✗ | ✓ |
| | XGBoost (OI) | ✗ | ✓ | ✓ | ✓ | ✗ | ✓ | ✓ | ✓ | ✗ | ✓ | ✓ | ✓ |
| | SVM (OI) | ✓ | ✓ | ✓ | ✓ | ✗ | ✗ | ✓ | ✓ | ✗ | ✗ | ✗ | ✓ |
| | EM (OI) | ✗ | ✗ | ✗ | ✗ | ✗ | ✓ | ✗ | ✗ | ✓ | ✗ | ✗ | ✗ |
| MAE | KNN (OI) | ✓ | ✓ | ✓ | ✓ | ✗ | ✗ | ✗ | ✗ | ✓ | ✗ | ✓ | ✗ |
| | MissForest (OI) | ✓ | ✗ | ✓ | ✓ | ✗ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| | XGBoost (OI) | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| | SVM (OI) | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✗ | ✓ | ✗ |
| | EM (OI) | ✗ | ✓ | ✓ | ✓ | ✗ | ✓ | ✓ | ✓ | ✗ | ✗ | ✗ | ✗ |
| Wasserstein dist. | KNN (OI) | ✓ | ✓ | ✓ | ✓ | ✗ | ✗ | ✗ | ✗ | ✗ | ✓ | ✗ | ✓ |
| | MissForest (OI) | ✗ | ✗ | ✗ | ✗ | ✗ | ✗ | ✗ | ✗ | ✓ | ✓ | ✗ | ✓ |
| | XGBoost (OI) | ✓ | ✗ | ✓ | ✓ | ✗ | ✓ | ✗ | ✓ | ✓ | ✓ | ✓ | ✓ |
| | SVM (OI) | ✗ | ✗ | ✗ | ✓ | ✗ | ✗ | ✓ | ✓ | ✗ | ✗ | ✗ | ✗ |
| | EM (OI) | ✓ | ✗ | ✗ | ✗ | ✓ | ✗ | ✗ | ✗ | ✗ | ✗ | ✗ | ✗ |

* ✓ indicates that adding the outlier indicator variable improves the performance; ✗ indicates no improvement or degraded the performance.

- **The impact of incorporating an outlier indicator for n = 200 and k = 20:**
  Table 4 presents results for the impact of adding an outlier indicator variable for different imputation methods across different missing and contamination rates for n = 200 and k = 20 according to RMSE, MAE and Wasserstein distance. The following key insights can be drawn:
  - For RMSE results, incorporating an outlier indicator improves RMSE performance for XGBoost for all contamination levels except the ones with low missing rates. Also, it enhances SVM results at low contamination levels. In addition to KNN for moderate (10% and 20%) and high missing rates at 5% contamination. On the other hand, not incorporating it is better for EM in most scenarios.
  - According to MAE, adding an outlier indicator enhances XGBoost results for all contamination levels under different missingness conditions. SVM and EM also demonstrated better results at low and moderate contamination levels under all missing rates. MissForest, for all contamination levels and most missing rates, benefits from adding an outlier indicator.
  - For Wasserstein distance performance, outlier indicator incorporating improves KNN results at low (5%) contamination across all missing rates, and some missing rates for high (20%) contamination. In addition to improving MissForest results for high contamination (except 20% missingness). XGBoost also benefits from the outlier indicator according to Wasserstein distance for most scenarios almost at moderate and high missing rates across different contamination levels. Also, SVM shows better performance at 5% and 10% contamination with high missing rates. In contrast, it doesn't improve EM results in most scenarios.

- **Overall performance for n = 500 and k = 20:**
  Tables 5, 6, and 7 present the simulated RMSE, MAE, and Wasserstein distance for different imputation methods across different missing rates and contamination levels for n = 500 and k = 20. Key insights can be concluded as follows:
  - The proposed RESF-IM outperforms other algorithms for most scenarios according to all evaluation metrics. The proposed RKSF-IM and MissForest come after.
  - RMSE results confirm the usual results where the two proposed methods outperform all other algorithms.

Table 5. Simulated RMSE for different imputation methods across different missing rates with 5%, 10%, and 20% contamination for n = 500 and k = 20.

| | 5% contamination | | | | 10% contamination | | | | 20% contamination | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Missing rate | 5% | 10% | 20% | 30% | 5% | 10% | 20% | 30% | 5% | 10% | 20% | 30% |
| Mean | 0.1922 | 0.2991 | 0.4379 | 0.5387 | 0.1996 | 0.2980 | 0.4325 | 0.5409 | 0.1860 | 0.2970 | 0.4439 | 0.5440 |
| Median | 0.1922 | 0.2990 | 0.4378 | 0.5386 | 0.1992 | 0.2978 | 0.4324 | 0.5406 | 0.1859 | 0.2968 | 0.4435 | 0.5436 |
| KNN (OI) | 0.1404 | 0.2267 | 0.3513 | 0.4434 | 0.1435 | 0.2232 | 0.3455 | 0.4476 | 0.1373 | 0.2211 | 0.3475 | 0.4439 |
| KNN (WOI) | 0.1392 | 0.2267 | 0.3495 | 0.4426 | 0.1453 | 0.2269 | 0.3471 | 0.4503 | 0.1375 | 0.2216 | 0.3497 | 0.4479 |
| IRMI | 0.1567 | 0.2356 | 0.3383 | 0.4637 | 0.1518 | 0.2281 | 0.3275 | 0.4129 | 0.1407 | 0.2188 | 0.3273 | 0.4958 |
| MissForest (OI) | 0.1317 | 0.2069 | 0.3217 | 0.4081 | 0.1346 | 0.2095 | 0.3169 | 0.4064 | 0.1286 | 0.2059 | 0.3186 | 0.4048 |
| MissForest (WOI) | 0.1319 | 0.2081 | 0.3225 | 0.4101 | 0.1353 | 0.2108 | 0.3183 | 0.4096 | 0.1283 | 0.2061 | 0.3211 | 0.4083 |
| XGBoost (OI) | 0.1377 | 0.2178 | 0.3396 | 0.4334 | 0.1424 | 0.2225 | 0.3364 | 0.4343 | 0.1351 | 0.2177 | 0.3381 | 0.4302 |
| XGBoost (WOI) | 0.1378 | 0.2164 | 0.3415 | 0.4375 | 0.1424 | 0.2250 | 0.3420 | 0.4419 | 0.1348 | 0.2183 | 0.3432 | 0.4401 |
| SVM (OI) | 0.1457 | 0.2428 | 0.3774 | 0.4783 | 0.1585 | 0.2519 | 0.3833 | 0.4951 | 0.1432 | 0.2463 | 0.3907 | 0.4983 |
| SVM (WOI) | 0.1450 | 0.2420 | 0.3768 | 0.4789 | 0.1600 | 0.2521 | 0.3833 | 0.4949 | 0.1426 | 0.2454 | 0.3902 | 0.4980 |
| EM (OI) | 0.1839 | 0.2775 | 0.4182 | 0.5234 | 0.1854 | 0.2800 | 0.4136 | 0.5252 | 0.1808 | 0.2802 | 0.4181 | 0.5253 |
| EM (WOI) | 0.1829 | 0.2789 | 0.4170 | 0.5249 | 0.1874 | 0.2825 | 0.4149 | 0.5252 | 0.1851 | 0.2795 | 0.4179 | 0.5265 |
| RESF-IM (Xgb) | 0.1243 | 0.1963 | 0.3059 | 0.3909 | 0.1276 | 0.1987 | 0.3020 | 0.3929 | 0.1211 | 0.1949 | 0.3068 | 0.4063 |
| RESF-IM (EM) | 0.1244 | 0.1953 | 0.3046 | 0.3880 | 0.1277 | 0.1978 | 0.3007 | 0.3900 | 0.1212 | 0.1948 | 0.3060 | 0.4041 |
| RKSF-IM (EM) | 0.1316 | 0.2057 | 0.3176 | 0.4050 | 0.1342 | 0.2067 | 0.3132 | 0.4074 | 0.1257 | 0.2033 | 0.3174 | 0.4065 |
| RKSF-IM (Xgb) | 0.1326 | 0.2059 | 0.3190 | 0.4078 | 0.1330 | 0.2071 | 0.3148 | 0.4095 | 0.1272 | 0.2031 | 0.3191 | 0.4091 |

*OI indicates with outlier indicator, WOI indicates without outlier indicator.

🟩 First and second ranks 🟨 Third and fourth ranks 🟧 Fifth and sixth ranks

Table 6. Simulated MAE for different imputation methods across different missing rates with 5%, 10%, and 20% contamination for n = 500 and k = 20.

| | 5% contamination | | | | 10% contamination | | | | 20% contamination | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Missing rate | 5% | 10% | 20% | 30% | 5% | 10% | 20% | 30% | 5% | 10% | 20% | 30% |
| Mean | 0.0345 | 0.0678 | 0.1344 | 0.2009 | 0.0341 | 0.0640 | 0.1261 | 0.1898 | 0.0314 | 0.0615 | 0.1245 | 0.1851 |
| Median | 0.0345 | 0.0678 | 0.1344 | 0.2007 | 0.0340 | 0.0639 | 0.1260 | 0.1896 | 0.0314 | 0.0615 | 0.1243 | 0.1849 |
| KNN (OI) | 0.0242 | 0.0445 | 0.1004 | 0.1550 | 0.0238 | 0.0457 | 0.0945 | 0.1479 | 0.0223 | 0.0442 | 0.0934 | 0.1449 |
| KNN (WOI) | 0.0241 | 0.0445 | 0.1002 | 0.1549 | 0.0241 | 0.0462 | 0.0948 | 0.1484 | 0.0224 | 0.0444 | 0.0937 | 0.1458 |
| IRMI | 0.0283 | 0.0542 | 0.1052 | 0.1586 | 0.0265 | 0.0503 | 0.0977 | 0.1456 | 0.0245 | 0.0475 | 0.0952 | 0.1549 |
| MissForest (OI) | 0.0228 | 0.0449 | 0.0933 | 0.1452 | 0.0223 | 0.0431 | 0.0878 | 0.1372 | 0.0210 | 0.0414 | 0.0863 | 0.1341 |
| MissForest (WOI) | 0.0229 | 0.0452 | 0.0939 | 0.1464 | 0.0224 | 0.0434 | 0.0883 | 0.1385 | 0.0209 | 0.0415 | 0.0867 | 0.1352 |
| XGBoost (OI) | 0.0238 | 0.0474 | 0.0985 | 0.1529 | 0.0235 | 0.0457 | 0.0929 | 0.1455 | 0.0221 | 0.0437 | 0.0917 | 0.1427 |
| XGBoost (WOI) | 0.0239 | 0.0473 | 0.0993 | 0.1556 | 0.0235 | 0.0459 | 0.0943 | 0.1484 | 0.0220 | 0.0439 | 0.0927 | 0.1460 |
| SVM (OI) | 0.0242 | 0.0487 | 0.1014 | 0.1588 | 0.0247 | 0.0476 | 0.0978 | 0.1543 | 0.0225 | 0.0462 | 0.0986 | 0.1540 |
| SVM (WOI) | 0.0242 | 0.0487 | 0.1017 | 0.1600 | 0.0249 | 0.0477 | 0.0981 | 0.1549 | 0.0224 | 0.0461 | 0.0986 | 0.1543 |
| EM (OI) | 0.0335 | 0.0652 | 0.1350 | 0.2077 | 0.0332 | 0.0651 | 0.1331 | 0.2065 | 0.0331 | 0.0660 | 0.1358 | 0.2076 |
| EM (WOI) | 0.0335 | 0.0661 | 0.1362 | 0.2094 | 0.0337 | 0.0659 | 0.1344 | 0.2080 | 0.0336 | 0.0654 | 0.1357 | 0.2087 |
| RESF-IM (Xgb) | 0.0216 | 0.0430 | 0.0903 | 0.1411 | 0.0213 | 0.0411 | 0.0846 | 0.1335 | 0.0199 | 0.0393 | 0.0831 | 0.1323 |
| RESF-IM (EM) | 0.0216 | 0.0429 | 0.0899 | 0.1402 | 0.0213 | 0.0410 | 0.0842 | 0.1325 | 0.0199 | 0.0392 | 0.0828 | 0.1315 |
| RKSF-IM (EM) | 0.0230 | 0.0454 | 0.0939 | 0.1455 | 0.0225 | 0.0433 | 0.0887 | 0.1385 | 0.0208 | 0.0417 | 0.0871 | 0.1356 |
| RKSF-IM (Xgb) | 0.0230 | 0.0453 | 0.0937 | 0.1455 | 0.0225 | 0.0433 | 0.0884 | 0.1382 | 0.0210 | 0.0414 | 0.0869 | 0.1355 |

*OI indicates with outlier indicator, WOI indicates without outlier indicator.

🟩 First and second ranks 🟨 Third and fourth ranks 🟧 Fifth and sixth ranks

- For MAE, MissForest with outlier indicator performs better than RKSF-IM in most scenarios.
- According to the Wasserstein distance, RKSF-IM is almost the second-best method at all contamination levels, except for high missing rates, where MissForest and IRMI come before it.

- **The impact of incorporating an outlier indicator for n = 500 and k = 20:**
  Table 8 presents results for the impact of adding an outlier indicator variable for different imputation methods across different missing and contamination rates for n = 500 and k = 20 according to RMSE, MAE and Wasserstein distance. The following key insights can be drawn:

Table 7. Simulated Wasserstein distance for different imputation methods across different missing rates with 5%, 10%, and 20% contamination for n = 500 and k = 20.

| | 5% contamination | | | | 10% contamination | | | | 20% contamination | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Missing rate | 5% | 10% | 20% | 30% | 5% | 10% | 20% | 30% | 5% | 10% | 20% | 30% |
| Mean | 0.0390 | 0.0556 | 0.0711 | 0.0932 | 0.0413 | 0.0580 | 0.0806 | 0.0974 | 0.0471 | 0.0661 | 0.0911 | 0.1094 |
| Median | 0.0390 | 0.0556 | 0.0711 | 0.0932 | 0.0413 | 0.0580 | 0.0806 | 0.0974 | 0.0471 | 0.0660 | 0.0910 | 0.1093 |
| KNN (OI) | 0.0299 | 0.0437 | 0.0618 | 0.0760 | 0.0312 | 0.0447 | 0.0651 | 0.0803 | 0.0354 | 0.0503 | 0.0723 | 0.0892 |
| KNN (WOI) | 0.0294 | 0.0432 | 0.0616 | 0.0760 | 0.0313 | 0.0451 | 0.0653 | 0.0807 | 0.0350 | 0.0501 | 0.0724 | 0.0896 |
| IRMI | 0.0300 | 0.0416 | 0.0563 | 0.0680 | 0.0307 | 0.0426 | 0.0589 | 0.0715 | 0.0343 | 0.0477 | 0.0661 | 0.0818 |
| MissForest (OI) | 0.0274 | 0.0401 | 0.0567 | 0.0700 | 0.0292 | 0.0415 | 0.0596 | 0.0727 | 0.0330 | 0.0466 | 0.0661 | 0.0805 |
| MissForest (WOI) | 0.0273 | 0.0401 | 0.0567 | 0.0701 | 0.0292 | 0.0416 | 0.0597 | 0.0728 | 0.0331 | 0.0466 | 0.0661 | 0.0804 |
| XGBoost (OI) | 0.0285 | 0.0418 | 0.0591 | 0.0733 | 0.0306 | 0.0436 | 0.0621 | 0.0767 | 0.0346 | 0.0487 | 0.0688 | 0.0835 |
| XGBoost (WOI) | 0.0285 | 0.0420 | 0.0597 | 0.0738 | 0.0307 | 0.0438 | 0.0628 | 0.0769 | 0.0347 | 0.0491 | 0.0693 | 0.0843 |
| SVM (OI) | 0.0323 | 0.0473 | 0.0674 | 0.0830 | 0.0351 | 0.0506 | 0.0726 | 0.0894 | 0.0388 | 0.0539 | 0.0808 | 0.1002 |
| SVM (WOI) | 0.0322 | 0.0471 | 0.0672 | 0.0830 | 0.0352 | 0.0506 | 0.0725 | 0.0893 | 0.0386 | 0.0536 | 0.0805 | 0.1000 |
| EM (OI) | 0.0335 | 0.0472 | 0.0646 | 0.0774 | 0.0356 | 0.0499 | 0.0685 | 0.0820 | 0.0406 | 0.0564 | 0.0772 | 0.0917 |
| EM (WOI) | 0.0334 | 0.0470 | 0.0642 | 0.0768 | 0.0355 | 0.0495 | 0.0683 | 0.0817 | 0.0407 | 0.0560 | 0.0771 | 0.0916 |
| RESF-IM (Xgb) | 0.0257 | 0.0377 | 0.0545 | 0.0684 | 0.0274 | 0.0394 | 0.0574 | 0.0715 | 0.0314 | 0.0447 | 0.0644 | 0.0801 |
| RESF-IM (EM) | 0.0256 | 0.0376 | 0.0541 | 0.0665 | 0.0273 | 0.0392 | 0.0570 | 0.0706 | 0.0313 | 0.0444 | 0.0641 | 0.0794 |
| RKSF-IM (EM) | 0.0267 | 0.0394 | 0.0561 | 0.0702 | 0.0286 | 0.0409 | 0.0591 | 0.0735 | 0.0326 | 0.0462 | 0.0661 | 0.0817 |
| RKSF-IM (Xgb) | 0.0269 | 0.0396 | 0.0565 | 0.0707 | 0.0287 | 0.0411 | 0.0595 | 0.0737 | 0.0327 | 0.0463 | 0.0663 | 0.0829 |

*OI indicates with outlier indicator, WOI indicates without outlier indicator.

🟩 First and second ranks 🟨 Third and fourth ranks 🟧 Fifth and sixth ranks

Table 8. Simulation study results for the impact of adding an outlier indicator variable for different imputation methods across different missing and contamination rates according to RMSE, MAE and Wasserstein distance for n = 500 and k = 20.

| | | 5% contamination | | | | 10% contamination | | | | 20% contamination | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Missing rate | 5% | 10% | 20% | 30% | 5% | 10% | 20% | 30% | 5% | 10% | 20% | 30% |
| RMSE | KNN (OI) | ✗ | ✓ | ✗ | ✗ | ✓ | ✓ | ✓ | ✗ | ✓ | ✓ | ✓ | ✓ |
| | MissForest (OI) | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✗ | ✓ | ✓ | ✓ |
| | XGBoost (OI) | ✓ | ✗ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✗ | ✓ | ✓ | ✓ |
| | SVM (OI) | ✗ | ✗ | ✗ | ✓ | ✓ | ✓ | ✓ | ✗ | ✗ | ✗ | ✗ | ✗ |
| | EM (OI) | ✗ | ✓ | ✗ | ✓ | ✓ | ✓ | ✓ | ✗ | ✓ | ✗ | ✗ | ✓ |
| MAE | KNN (OI) | ✗ | ✓ | ✗ | ✗ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| | MissForest (OI) | ✗ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✗ | ✓ | ✓ | ✓ |
| | XGBoost (OI) | ✓ | ✗ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| | SVM (OI) | ✗ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✗ | ✗ | ✗ | ✓ |
| | EM (OI) | ✗ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✗ | ✗ | ✗ | ✓ |
| Wasserstein dist. | KNN (OI) | ✗ | ✗ | ✗ | ✗ | ✓ | ✓ | ✓ | ✓ | ✗ | ✗ | ✓ | ✓ |
| | MissForest (OI) | ✗ | ✗ | ✗ | ✓ | ✓ | ✓ | ✓ | ✓ | ✗ | ✗ | ✗ | ✗ |
| | XGBoost (OI) | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| | SVM (OI) | ✗ | ✗ | ✗ | ✗ | ✓ | ✓ | ✗ | ✗ | ✗ | ✗ | ✗ | ✗ |
| | EM (OI) | ✗ | ✗ | ✗ | ✗ | ✗ | ✗ | ✗ | ✗ | ✗ | ✗ | ✗ | ✗ |

* ✓ indicates that adding the outlier indicator variable improves the performance; ✗ indicates no improvement or degraded the performance.

- Incorporating an outlier indicator improves the performance of RMSE and MAE for MissForest and XGBoost in most scenarios.
- For Wasserstein distance performance, the outlier indicator incorruption improves XGBoost results at all scenarios, while it improves MissForest and KNN results only at 10% contamination. In contrast, it does not improve EM results in all scenarios, and SVM in most scenarios.

### 4.3. *Real data applications*

This subsection employs two different real-life applications with different correlation structures where the multivariate normality assumption is violated. The results support the superiority of the proposed frameworks in most scenarios. The first application is the males' blood pressure [18]. It includes 175 observations, 128 were tested negative (regular/normal) and 47 tested positive (hypertension) [23]. Five continuous variables are selected, which are present in Table 9. DBSCAN with minpts = 6 and eps = 8 detected 26 observations as outliers. This result is generally for the complete selected data, while it changes slightly depending on the missing rate. Shapiro-Wilk test statistic for multivariate normality is 0.857 with p-value $< 0.001$, assuring the violation of the multivariate normality. Figure 3 shows Spearman correlation with 95% bootstrapped confidence interval (1000 bootstraps).

Table 9. Descriptive statistics and outlier detection for males' blood pressure dataset.

| **Variable** | **Description** | **Median** | **Mean** | **SD** | **IQR** | **Shapiro-Wilk** | **p-value** | **N Out** | **Out%** |
|---|---|---|---|---|---|---|---|---|---|
| Age | Age in years | 22 | 23.81 | 6.71 | 6.5 | 0.752 | $< 0.001$ | 8 | 4.57 |
| bmi | Body mass index | 24.82 | 24.95 | 4.92 | 6.0 | 0.940 | $< 0.001$ | 5 | 2.86 |
| wc | Waist circumference (inch) | 84 | 86.06 | 11.47 | 14 | 0.959 | $< 0.001$ | 6 | 3.43 |
| hc | Hip circumference (inch) | 103 | 102.93 | 8.93 | 11 | 0.988 | 0.150 | 4 | 2.29 |
| whr | Waist hip ratio | 83 | 83.48 | 6.76 | 7 | 0.923 | $< 0.001$ | 5 | 2.86 |

*N Out = number of outliers, Out% = outlier percentage.

*Outliers are detected using the IQR method (lower bound = Q1 - 1.5 × IQR, upper bound = Q3 + 1.5 × IQR).

The second application employs the dataset provided by [46] which is available in the faraway R package with the name diabetes. The original dataset contains 403 instances. After cleaning the data to remove missing values, as we want the data to be complete to artificially introduce missing values and compare the imputed values with the original values, the dataset becomes 370 instances. Thirteen continuous variables are selected, which are present in Table 10. DBSCAN with minpts = 14 and eps = 130 detected 28 observations as outliers, this result is generally for the complete selected data, while it changes slightly depending on the missing rate. Shapiro-Wilk test statistic for multivariate normality is 0.904 with p-value $< 0.001$, indicating non-normality. Figure 4 shows the Spearman correlation heatmap. In general, the two datasets consider different correlation structures among the variables.

Table 10. Descriptive statistics and outlier detection for diabetes dataset.

| **Variable** | **Description** | **Median** | **Mean** | **SD** | **IQR** | **Shapiro-Wilk** | **p-value** | **N Out** | **Out%** |
|---|---|---|---|---|---|---|---|---|---|
| chol | Total Cholesterol (mg/dL) | 204 | 207.57 | 44.7 | 50.5 | 0.95 | $< 0.001$ | 10 | 2.67 |
| stab.glu | Stabilized Glucose (mg/dL) | 90 | 107.62 | 54.08 | 27.5 | 0.65 | $< 0.001$ | 48 | 12.8 |
| hdl | HDL (mg/dL) | 46 | 50.43 | 17.44 | 21 | 0.92 | $< 0.001$ | 13 | 3.47 |
| ratio | Cholesterol/HDL Ratio | 4.2 | 4.53 | 1.76 | 2.2 | 0.86 | $< 0.001$ | 6 | 1.6 |
| glyhb | Glycosylated hemoglobin | 4.86 | 5.6 | 2.22 | 1.24 | 0.72 | $< 0.001$ | 53 | 14.13 |
| age | Age (years) | 45 | 46.98 | 16.66 | 26 | 0.97 | $< 0.001$ | 0 | 0 |
| height | Height (inches) | 66 | 66 | 3.92 | 6 | 0.99 | 0.003 | 1 | 0.27 |
| weight | Weight (pounds) | 174 | 177.89 | 40.57 | 49 | 0.96 | $< 0.001$ | 11 | 2.93 |
| bp.1s | First Systolic BP (mmHg) | 136 | 137.45 | 23.18 | 26.5 | 0.94 | $< 0.001$ | 12 | 3.2 |
| bp.1d | First Diastolic BP (mmHg) | 82 | 83.38 | 13.54 | 16 | 0.99 | 0.027 | 9 | 2.4 |
| waist | Waist (inches) | 37 | 37.95 | 5.78 | 8.5 | 0.98 | $< 0.001$ | 2 | 0.53 |
| hip | Hip (inches) | 42 | 43.09 | 5.64 | 7 | 0.96 | $< 0.001$ | 12 | 3.2 |
| time.ppn | Postprandial time (min) | 240 | 335.01 | 309.06 | 360 | 0.86 | $< 0.001$ | 10 | 2.67 |

*N Out = number of outliers, Out% = outlier percentage.

*Outliers are detected using the IQR method (lower bound = Q1 - 1.5 × IQR, upper bound = Q3 + 1.5 × IQR).

### 4.4. *Real data results*

Table 11 shows the results of the performance metrics for the males' blood pressure dataset. Several key insights can be concluded as follows:

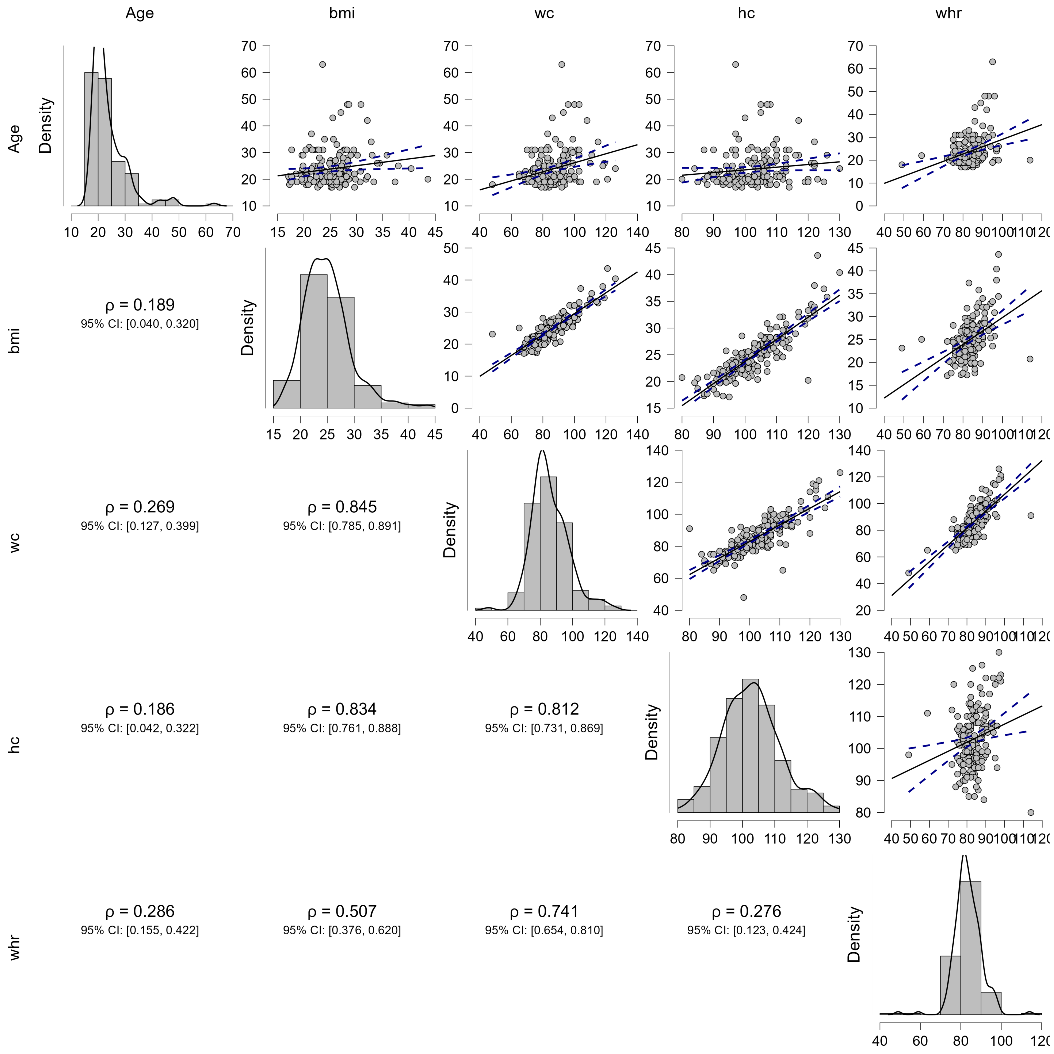

Figure 3. Spearman correlation with 95% bootstrapped confidence interval (1000 bootstraps) for males' blood pressure dataset.

- For RMSE, RESF-IM with its two versions almost has the best performance across different missing rates (except for 20% missing rate) and RKSF-IM with its two versions comes next. While the median and the mean have the lowest performance.
- For MAE, RESF-IM and RKSF-IM, are almost in the top three methods.
- The effectiveness of adding outlier indicators shows an improvement in the results for KNN, MissForest, and EM at low (5%) and moderate missing rates (10% and 20%) in all metrics. However, as missing rates increase to 30%, the deterioration in the effectiveness of the outlier indicator is particularly evident in the RMSE and MAE metrics, where these three methods show better performance without adding an outlier indicator.
- For XGBoost adding an outlier indicator shows better performance at 20% and 30% missing rates according to RMSE and MAE. This result is reversed for the Wasserstein distance, which shows better performance for adding outlier indicator at low and moderate (5%, 10%, and 20%) missing rates.

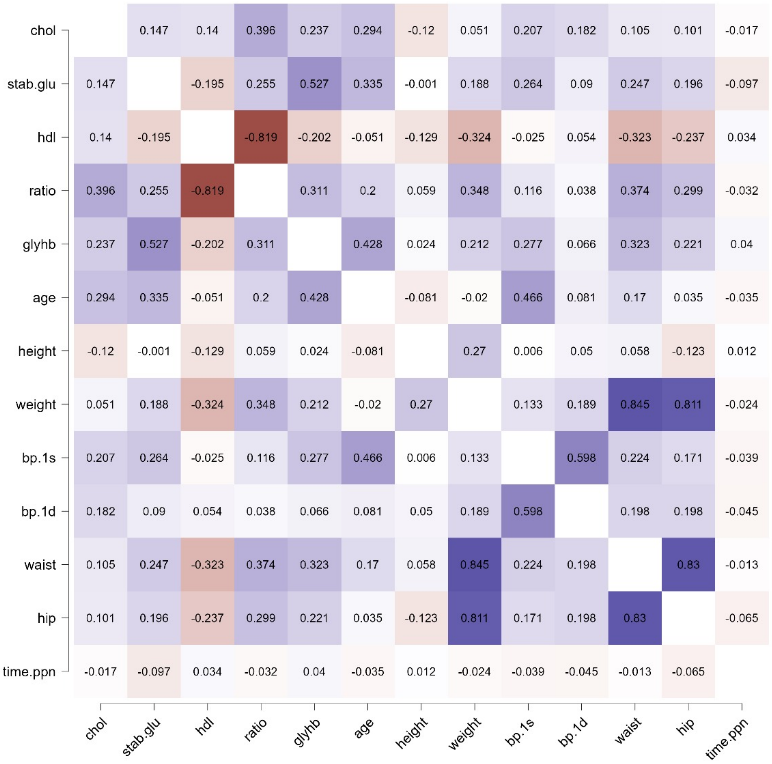

Figure 4. Spearman correlation heatmap for diabetes dataset.

Table 11. RMSE, MAE, Wasserstein distance for different imputation methods across 5%, 10%, 20% and 30% missing rates for males' blood pressure dataset.

| Missing rate | RMSE | | | | MAE | | | | Wasserstein dist | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | 5% | 10% | 20% | 30% | 5% | 10% | 20% | 30% | 5% | 10% | 20% | 30% |
| Median | 1.146 | 1.178 | 1.100 | 1.058 | 0.817 | 0.851 | 0.787 | 0.750 | 0.078 | 0.103 | 0.130 | 0.153 |
| Mean | 1.118 | 1.159 | 1.082 | 1.035 | 0.816 | 0.853 | 0.791 | 0.764 | 0.077 | 0.103 | 0.129 | 0.154 |
| KNN (OI) | 0.692 | 0.750 | 0.877 | 0.894 | 0.475 | 0.429 | 0.578 | 0.586 | 0.045 | 0.063 | 0.094 | 0.113 |
| KNN (WOI) | 0.708 | 0.762 | 0.888 | 0.879 | 0.494 | 0.453 | 0.588 | 0.576 | 0.046 | 0.067 | 0.097 | 0.116 |
| IRMI | 0.747 | 0.634 | 0.733 | 0.860 | 0.382 | 0.356 | 0.471 | 0.482 | 0.036 | 0.049 | 0.073 | 0.089 |
| MissForest (OI) | 0.602 | 0.691 | 0.735 | 0.784 | 0.402 | 0.463 | 0.477 | 0.507 | 0.039 | 0.058 | 0.082 | 0.095 |
| MissForest (WOI) | 0.656 | 0.749 | 0.800 | 0.744 | 0.404 | 0.472 | 0.522 | 0.487 | 0.040 | 0.063 | 0.083 | 0.096 |
| XGBoost (OI) | 0.628 | 0.644 | 0.776 | 0.790 | 0.410 | 0.416 | 0.502 | 0.501 | 0.040 | 0.055 | 0.084 | 0.095 |
| XGBoost (WOI) | 0.585 | 0.628 | 0.842 | 0.797 | 0.404 | 0.401 | 0.541 | 0.523 | 0.041 | 0.056 | 0.087 | 0.094 |
| SVM (OI) | 0.659 | 0.827 | 0.852 | 0.876 | 0.354 | 0.418 | 0.527 | 0.580 | 0.043 | 0.067 | 0.099 | 0.125 |
| SVM (WOI) | 0.664 | 0.759 | 0.832 | 0.895 | 0.359 | 0.393 | 0.531 | 0.582 | 0.044 | 0.065 | 0.091 | 0.128 |
| EM (OI) | 0.558 | 0.683 | 0.768 | 0.889 | 0.338 | 0.370 | 0.486 | 0.574 | 0.036 | 0.046 | 0.068 | 0.088 |
| EM (WOI) | 0.790 | 0.839 | 0.782 | 0.872 | 0.429 | 0.504 | 0.503 | 0.563 | 0.036 | 0.051 | 0.070 | 0.091 |
| RESF-IM (EM) | 0.543 | 0.551 | 0.715 | 0.722 | 0.295 | 0.348 | 0.467 | 0.453 | 0.032 | 0.048 | 0.078 | 0.093 |
| RESF-IM (Xgb) | 0.543 | 0.566 | 0.700 | 0.715 | 0.306 | 0.363 | 0.471 | 0.445 | 0.033 | 0.049 | 0.080 | 0.095 |
| RKSF-IM (EM) | 0.578 | 0.596 | 0.713 | 0.767 | 0.317 | 0.393 | 0.468 | 0.512 | 0.033 | 0.050 | 0.078 | 0.100 |
| RKSF-IM (Xgb) | 0.567 | 0.611 | 0.736 | 0.759 | 0.284 | 0.379 | 0.486 | 0.490 | 0.033 | 0.051 | 0.081 | 0.097 |

*OI indicates with outlier indicator, WOI indicates without outlier indicator.

🟩 First and second ranks 🟨 Third and fourth ranks 🟧 Fifth and sixth ranks

- SVM with outlier indicator shows good performance in low and high missing rates for RMSE and Wasserstein distance.
- In general, the Wasserstein distance metric, which captures distributional differences, shows more consistent benefits from adding an outlier indicator across different methods and missing rates.
- Including an outlier indicator shows a good improvement in general for RMSE. It performs better in 13 out of 20 scenarios (Five imputation methods with and without adding an outlier indicator across four missing rates) compared to only seven scenarios without the outlier indicator. This represents a 65% success rate when an outlier indicator is included. Approximately, similar results are observed for MAE, where including an outlier indicator enhances performance in 14 out of 20 scenarios (70% success rate) versus six scenarios without an outlier indicator. Finally, for the Wasserstein distance, adding an outlier indicator wins in 16 scenarios (80%) versus four scenarios without an outlier indicator.

Table 12. RMSE, MAE, Wasserstein distance for different imputation methods across 5%, 10%, 20% and 30% missing rates for diabetes dataset.

| Missing rate | RMSE | | | | MAE | | | | Wasserstein dist | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | 5% | 10% | 20% | 30% | 5% | 10% | 20% | 30% | 5% | 10% | 20% | 30% |
| Median | 0.988 | 0.973 | 1.022 | 1.022 | 0.688 | 0.712 | 0.735 | 0.735 | 0.134 | 0.188 | 0.269 | 0.269 |
| Mean | 0.960 | 0.941 | 1.001 | 1.001 | 0.711 | 0.724 | 0.762 | 0.762 | 0.130 | 0.182 | 0.265 | 0.265 |
| KNN (OI) | 0.840 | 0.886 | 0.948 | 0.948 | 0.556 | 0.628 | 0.680 | 0.680 | 0.113 | 0.169 | 0.247 | 0.247 |
| KNN (WOI) | 0.839 | 0.878 | 0.95 | 0.95 | 0.557 | 0.624 | 0.682 | 0.682 | 0.113 | 0.167 | 0.246 | 0.246 |
| IRMI | 0.886 | 0.872 | 0.851 | 0.851 | 0.623 | 0.613 | 0.585 | 0.585 | 0.117 | 0.166 | 🟧 0.21 | 🟧 0.21 |
| MissForest (OI) | 0.745 | 0.72 | 🟧 0.805 | 🟧 0.806 | 0.539 | 0.525 | 🟧 0.580 | 0.580 | 0.103 | 🟨 0.138 | 0.214 | 0.214 |
| MissForest (WOI) | 0.721 | 0.735 | 0.810 | 0.810 | 0.508 | 0.539 | 0.574 | 🟧 0.574 | 0.099 | 0.143 | 0.214 | 0.214 |
| XGBoost (OI) | 🟧 0.715 | 🟧 0.727 | 🟧 0.790 | 🟧 0.790 | 🟧 0.495 | 🟧 0.513 | 🟧 0.554 | 🟧 0.554 | 🟧 0.097 | 0.141 | 🟧 0.209 | 🟧 0.209 |
| XGBoost (WOI) | 🟧 0.7148 | 🟧 0.72 | 0.812 | 0.812 | 0.499 | 🟨 0.508 | 0.573 | 0.573 | 🟧 0.098 | 🟧 0.140 | 0.215 | 0.215 |
| SVM (OI) | 0.749 | 0.818 | 0.971 | 0.971 | 🟧 0.498 | 0.568 | 0.693 | 0.693 | 0.103 | 0.160 | 0.255 | 0.255 |
| SVM (WOI) | 0.745 | 0.791 | 0.968 | 0.968 | 0.50 | 0.559 | 0.691 | 0.691 | 0.102 | 0.155 | 0.254 | 0.254 |
| EM (OI) | 1.019 | 1.025 | 1.060 | 1.060 | 0.766 | 0.754 | 0.769 | 0.769 | 0.132 | 0.176 | 0.24 | 0.24 |
| EM (WOI) | 1.012 | 0.987 | 1.051 | 1.051 | 0.73 | 0.752 | 0.766 | 0.766 | 0.12 | 0.179 | 0.228 | 0.2278 |
| RESF-IM (EM) | 🟩 0.679 | 🟩 0.680 | 🟨 0.785 | 🟨 0.785 | 🟩 0.478 | 🟩 0.481 | 🟨 0.555 | 🟨 0.555 | 🟨 0.095 | 🟩 0.135 | 🟨 0.207 | 🟨 0.207 |
| RESF-IM (Xgb) | 🟨 0.685 | 🟨 0.695 | 🟩 0.771 | 🟩 0.771 | 🟨 0.481 | 🟩 0.488 | 🟩 0.543 | 🟩 0.543 | 🟨 0.094 | 🟨 0.137 | 🟩 0.205 | 🟩 0.205 |
| RKSF-IM (EM) | 🟩 0.677 | 🟨 0.714 | 🟨 0.774 | 🟨 0.774 | 🟨 0.480 | 🟧 0.514 | 🟨 0.545 | 🟨 0.545 | 🟩 0.092 | 🟧 0.140 | 🟩 0.206 | 🟨 0.206 |
| RKSF-IM (Xgb) | 🟨 0.684 | 🟩 0.692 | 🟩 0.766 | 🟩 0.766 | 🟩 0.475 | 🟨 0.487 | 🟩 0.537 | 🟩 0.537 | 🟩 0.093 | 🟩 0.136 | 🟨 0.205 | 🟩 0.205 |

*OI indicates with outlier indicator, WOI indicates without outlier indicator.

🟩 First and second ranks 🟨 Third and fourth ranks 🟧 Fifth and sixth ranks

Table 12 shows the results of the performance metrics for the diabetes dataset. Several key insights can be drawn as follows:

- The proposed methods have the best performance across different missing rates and different performance metrics.
- The effectiveness of adding outlier indicators shows an enhancement in the results for MissForest across all metrics and for XGBoost at 20% and 30% missing rates.
- For EM, adding an outlier indicator shows better performance according to Wasserstein distance. At the same time, the results for RMSE and MAE are reversed, showing better performance without adding the outlier indicator.

## 5. Conclusion, limitations, and future work

This paper introduces two novel stacked ensemble frameworks, RSSF-IM and RESF-IM, both of which can effectively impute missing values for contaminated datasets. The paper also investigates the effectiveness of adding an outlier indicator to existing imputation methods to enhance their robustness. RMSE, MAE, and Wasserstein distances are evaluated across both simulated and real-world datasets with varying contamination levels (5%, 10%, and 20%) and missingness rates (5%, 10%, 20%, and 30%). The findings demonstrate that the proposed stacking-based approaches have the best performance over state-of-the-art imputation methods across all evaluation metrics and experimental settings. Results show that RESF-IM with its two variants achieve the lowest RMSE, MAE, and Wasserstein values across all scenarios involving both outliers and missingness. This suggests that the RESF-IM approach effectively combines the strengths of multiple base learners while eliminating their weaknesses. The RKSF-IM model, which represents an adaptation for the formal stacking technique for imputing missing values, ranks second in performance in most scenarios, further validating the strength of stacking-based frameworks in this domain. It is also observed that the performance decreases as the missing rate increases across all metrics, as expected. However, the Wasserstein distance results indicate that distributional similarity is better preserved than point estimates under high missingness scenarios. Among existing imputation methods, MissForest shows better performance than other competing approaches, while simple statistic methods such as mean or median imputation perform poorly in all scenarios. Adding an outlier indicator is useful for most of the methods at low and moderate contamination levels. In summary, under the simulation settings, our proposed stacking-based methods offer an accurate, robust and distribution-preserving solution for missing data imputation in the presence of outliers, providing a valuable advancement over traditional approaches.

Although results demonstrate the proposed frameworks' effectiveness, yet several limitations should be acknowledged. Firstly, computational efficiency represents a limitation of the proposed frameworks, as ensemble techniques inherently require additional computational resources due to the computational cost of multiple base learners, which represents a scalability challenge for high-dimensional data. Secondly, the proposed frameworks are constructed using DBSCAN for outlier detection. Further studies shall investigate other outlier detection methods with parameter sensitivity analysis. Thirdly, the simulation study assumed a contaminated MVN and MCAR missingness mechanism, whereas real-world applications involve more complex structures. Evaluating data under other distributions, rather than the MVN, for mixed data types, is recommended to provide comprehensive evidence of the framework's robustness. Additionally, evaluating performance under MAR and MNAR mechanisms shall also be addressed in the future work. Also, comparison against robust variants of other ensemble techniques (such as robust boosting or bagging methods) represents another limitation that needs future investigation. Future research directions also include adapting the stacking framework for multiple imputation contexts, testing different meta-learners to establish the framework's robustness across various configurations. Developing self-optimizing techniques for automatic selection of base learners and meta-learners will also enhance the framework's applicability.

## REFERENCES

1. M. A. Abdel-Fattah, *On defining a jackknifed pooled ridge-Liu estimator in beta regression: nonlinear programming evidence*, Communications in Statistics - Theory and Methods, 1–25, 2025. https://doi.org/10.1080/03610926.2025.2544937
2. M. A. Abdel-Fattah, *Improved Liu-ridge-type estimates for the beta regression model*, Journal of Statistical Computation and Simulation, vol. 94, no. 16, pp. 3533–3554, 2024. https://doi.org/10.1080/00949655.2024.2396001
3. M. A. Abdel-Fattah, *On a new class of binomial ridge-type regression estimators*, Communications in Statistics - Simulation and Computation, vol. 51, no. 6, pp. 3272–3290, 2022. https://doi.org/10.1080/03610918.2019.1711409
4. M. A. Abdel-Fattah, *Proposed SUR-ridge estimators with application*, Paper presented at the 48th Annual Conference on Statistics, Computer Science and Operation Research, Cairo University, Giza, Egypt, December 2013.
5. N. Abdulwahed, G. S. El-Tawel, and M. A. Makhlouf, *An efficient machine learning framework for disease gene prediction in Parkinson's disease and bladder cancer*, Statistics, Optimization & Information Computing, 2025.
6. C. Agostinelli, A. Leung, V. J. Yohai, and R. H. Zamar, *Robust estimation of multivariate location and scatter in the presence of cellwise and casewise contamination*, Test, vol. 24, no. 3, pp. 441–461, 2015.
7. M. Al-Hashimi, H. H. Alawjar, and M. Alawjar, *Ensemble method for intervention analysis to predict the water resources of the Tigris River*, Statistics, Optimization & Information Computing, 2025.
8. A. Aleryani, W. Wang, and B. De La Iglesia, *Multiple imputation ensembles (MIE) for dealing with missing data*, SN Computer Science, vol. 1, no. 3, p. 134, 2020.
9. A. Alimadad and M. Salibian-Barrera, *An outlier-robust fit for generalized additive models with applications to disease outbreak detection*, Journal of the American Statistical Association, vol. 106, no. 494, pp. 719–731, 2011.
10. M. Alwateer, E. S. Atlam, M. M. Abd El-Raouf, O. A. Ghoneim, and I. Gad, *Missing data imputation: A comprehensive review*, Journal of Computer and Communications, vol. 12, no. 11, pp. 53–75, 2024.
11. A. Andiojaya and H. Demirhan, *A bagging algorithm for the imputation of missing values in time series*, Expert Systems with Applications, vol. 129, pp. 10–26, 2019.

12. L. J. Beesley and J. M. Taylor, *A stacked approach for chained equations multiple imputation incorporating the substantive model*, Biometrics, vol. 77, no. 4, pp. 1342–1354, 2021.
13. A. P. Dempster, N. M. Laird, and D. B. Rubin, *Maximum likelihood from incomplete data via the EM algorithm*, Journal of the Royal Statistical Society: Series B (Methodological), vol. 39, no. 1, pp. 1–22, 1977.
14. S. Džeroski and B. Ženko, *Is combining classifiers with stacking better than selecting the best one?*, Machine Learning, vol. 54, no. 3, pp. 255–273, 2004.
15. B. Efron and R. J. Tibshirani, *An introduction to the bootstrap*, Chapman and Hall/CRC, 1994.
16. J. Elith, J. R. Leathwick, and T. Hastie, *A working guide to boosted regression trees*, Journal of Animal Ecology, vol. 77, no. 4, pp. 802–813, 2008.
17. T. Emmanuel, T. Maupong, D. Mpoeleng, T. Semong, B. Mphago, and O. Tabona, *A survey on missing data in machine learning*, Journal of Big Data, vol. 8, pp. 1–37, 2021.
18. H. F. Golino, L. S. D. B. Amaral, S. F. P. Duarte, C. M. A. Gomes, T. D. J. Soares, L. A. D. Reis, and J. Santos, *Predicting increased blood pressure using machine learning*, Journal of Obesity, vol. 2014, Article 637635, 2014.
19. L. K. Hansen and P. Salamon, *Neural network ensembles*, IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 12, no. 10, pp. 993–1001, 1990.
20. D. Harrison and D. L. Rubinfeld, *Hedonic prices and the demand for clean air*, Journal of Environmental Economics and Management, vol. 5, pp. 81–102, 1978.
21. M. K. Hasan, M. A. Alam, S. Roy, A. Dutta, M. T. Jawad, and S. Das, *Missing value imputation affects the performance of machine learning: A review and analysis of the literature (2010–2021)*, Informatics in Medicine Unlocked, vol. 27, Article 100799, 2021.
22. A. E. Hoerl and R. W. Kennard, *Ridge Regression: Biased Estimation for Nonorthogonal Problems*, Technometrics, **12**(1), 55–67, 1970.
23. M. F. Ijaz, G. Alfian, M. Syafrudin, and J. Rhee, *Hybrid prediction model for type 2 diabetes and hypertension using DBSCAN-based outlier detection, synthetic minority over-sampling technique (SMOTE), and random forest*, Applied Sciences, vol. 8, no. 8, Article 1325, 2018.
24. S. Kumar, M. K. Pandey, A. Nath, and K. Subbiah, *Performance analysis of ensemble supervised machine learning algorithms for missing value imputation*, in 2016 2nd International Conference on Computational Intelligence and Networks (CINE), IEEE, 2016, pp. 160–165.
25. R. J. Little and D. B. Rubin, *Statistical analysis with missing data*, John Wiley & Sons, 2019.
26. R. J. Little and N. Schenker, *Missing data*, in G. Arminger, C. C. Clogg, and M. E. Sobel (Eds.), Handbook of statistical modeling for the social and behavioral sciences, pp. 39–75, Springer, 1995.
27. G. Madhu, B. L. Bharadwaj, G. Nagachandrika, and K. S. Vardhan, *A novel algorithm for missing data imputation on machine learning*, in 2019 International Conference on Smart Systems and Inventive Technology (ICSSIT), IEEE, 2019, pp. 173–177.
28. M. N. Maulana, M. Muljono, and E. P. A. Meindiawan, *Comparative analysis of homogeneous and heterogeneous ensembles for diabetes classification optimization*, Sinkron: Jurnal dan Penelitian Teknik Informatika, vol. 9, no. 1, pp. 512–521, 2025.
29. G. Molenberghs, G. Fitzmaurice, M. G. Kenward, A. A. Tsiatis, and G. Verbeke (Eds.), *Handbook of missing data methodology*, CRC Press, 2015.
30. M. N. Noor, A. S. Yahaya, N. A. Ramli, and A. M. Al Bakri, *Mean imputation techniques for filling the missing observations in air pollution dataset*, Key Engineering Materials, vol. 594, pp. 902–908, 2014.
31. V. M. Panaretos and Y. Zemel, *An introduction to Wasserstein distances*, in Statistics and Computing, pp. 1–25, Springer, 2019.
32. K. I. Penny and I. T. Jolliffe, *A comparison of multivariate outlier detection methods for clinical laboratory safety data*, Journal of the Royal Statistical Society: Series D (The Statistician), vol. 50, no. 3, pp. 295–307, 2001.
33. B. Ramosaj and M. Pauly, *Predicting missing values: A comparative study on non-parametric approaches for imputation*, Computational Statistics, vol. 34, no. 4, pp. 1741–1764, 2019.
34. R. S. Rao, L. R. Kalabarige, B. Alankar, and A. K. Sahu, *Multimodal imputation-based stacked ensemble for prediction and classification of air quality index in Indian cities*, Computers and Electrical Engineering, vol. 114, Article 109098, 2024.
35. O. Sagi and L. Rokach, *Ensemble learning: A survey*, Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery, vol. 8, no. 4, Article e1249, 2018.
36. S. Serneels and T. Verdonck, *Principal component analysis for data containing outliers and missing elements*, Computational Statistics & Data Analysis, vol. 52, no. 3, pp. 1712–1727, 2008.
37. D. J. Stekhoven and P. Bühlmann, *MissForest—non-parametric missing value imputation for mixed-type data*, Bioinformatics, vol. 28, no. 1, pp. 112–118, 2012.
38. N. Tang and Y. Ju, *Statistical inference for nonignorable missing-data problems: A selective review*, Statistical Theory and Related Fields, vol. 2, no. 2, pp. 105–133, 2018.
39. M. Templ, *Enhancing precision in large-scale data analysis: An innovative robust imputation algorithm for managing outliers and missing values*, Mathematics, vol. 11, no. 12, Article 2729, 2023.
40. M. Templ, A. Kowarik, and P. Filzmoser, *Iterative stepwise regression imputation using standard and robust methods*, Computational Statistics and Data Analysis, vol. 55, no. 10, pp. 2793–2806, 2011.
41. M. Thurow, F. Dumpert, B. Ramosaj, and M. Pauly, *Goodness (of fit) of imputation accuracy: The GoodImpact analysis*, arXiv preprint arXiv:2101.07532, 2021.
42. C. T. Tran, M. Zhang, P. Andreae, and B. Xue, *Bagging and feature selection for classification with incomplete data*, in Applications of Evolutionary Computation: 20th European Conference, EvoApplications 2017, pp. 471–486, Springer, 2017.
43. B. Twala and M. Cartwright, *Ensemble imputation methods for missing software engineering data*, in 11th IEEE International Software Metrics Symposium (METRICS'05), IEEE, 2005, p. 10.
44. Q. Wang and H. Lu, *A novel stacking ensemble learner for predicting residual strength of corroded pipelines*, npj Materials Degradation, vol. 8, no. 1, Article 87, 2024.
45. R. Wicklin, *Simulating data with SAS*, SAS Institute, 2013.

46. Willems, J. P., Saunders, J. T., Hunt, D. E., & Schorling, J. B. (1997). Prevalence of coronary heart disease risk factors among rural blacks: a community-based study. *Southern Medical Journal*, *90*(8), 814–820.
47. D. H. Wolpert, *Stacked generalization*, Neural Networks, vol. 5, no. 2, pp. 241–259, 1992.
48. W. Zhang, R. Li, J. Zhao, J. Wang, X. Meng, and Q. Li, *Miss-gradient boosting regression tree: A novel approach to imputing water treatment data*, Applied Intelligence, vol. 53, no. 19, pp. 22917–22937, 2023.
49. X. Zhang, C. Yan, C. Gao, B. A. Malin, and Y. Chen, *Predicting missing values in medical data via XGBoost regression*, Journal of Healthcare Informatics Research, vol. 4, pp. 383–394, 2020.
50. Y. Zhang, J. Ma, S. Liang, X. Li, and J. Liu, *A stacking ensemble algorithm for improving the biases of forest aboveground biomass estimations from multiple remotely sensed datasets*, GIScience and Remote Sensing, vol. 59, no. 1, pp. 234–249, 2022.
51. Z.-H. Zhou, *Ensemble methods: Foundations and algorithms*, CRC Press, 2012.
52. D. Zhou and T. Shi, *Statistical inference based on distances between empirical distributions with applications to AIRS level 3 data*, in Proceedings of the NASA Conference on Intelligent Data Understanding (CIDU), 2011.