Adapting tree-based multiple imputation methods for multi-level data? A simulation study

Ketevan Gurtskaia[a*], Jakob Schwerter [a*] and Philipp Doebler[a]

[a]Department of Statistics, TU Dortmund University, Dortmund, Germany.
[*]Shared first authorship

ARTICLE HISTORY
Compiled February 13, 2024

Abstract

This simulation study evaluates the effectiveness of multiple imputation (MI) techniques for multilevel data. It compares the performance of traditional Multiple Imputation by Chained Equations (MICE) with tree-based methods such as Chained Random Forests with Predictive Mean Matching and Extreme Gradient Boosting. Adapted versions that include dummy variables for cluster membership are also included for the tree-based methods. Methods are evaluated for coefficient estimation bias, statistical power, and type I error rates on simulated hierarchical data with different cluster sizes (25 and 50) and levels of missingness (10% and 50%). Coefficients are estimated using random intercept and random slope models. The results show that while MICE is preferred for accurate rejection rates, Extreme Gradient Boosting is advantageous for reducing bias. Furthermore, the study finds that bias levels are similar across different cluster sizes, but rejection rates tend to be less favorable with fewer clusters (lower power, higher type I error). In addition, the inclusion of cluster dummies in tree-based methods improves estimation for Level 1 variables, but is less effective for Level

Communicating author contact details: Jakob Schwerter (0000-0001-5818-2431), TU Dortmund University, Department of Statistics, Martin-Schmeisser-Weg 13, D-44227 Dortmund, Germany. Email: jakob.schwerter@tu-dortmund.de. Simulation code can be found at https://osf.io/8256p/?view_only=04c185b782f7464eba7e796bf36b7231.

2 variables. When data become too complex and MICE is too slow, extreme gradient boosting is a good alternative for hierarchical data.

KEYWORDS
multiple imputation; multi-level data; MICE; missRanger; mixgb

1. Introduction

The issue of missing data, historically overlooked due to limited understanding and computational capabilities, gained prominence in the early 1970s, a pivotal period marked by technological advances (Heymann, 2017; Schafer & Olsen, 1998). Rubin (1976) recognized the shortcomings of conventional methods and challenged the idea that the causes of missing data were unimportant. Building on this, Little and Rubin (2002) elaborated on the treatment of missingness indicators as random variables and discussed contemporary methods for dealing with the problem. This paper examines new technological advances in the area of tree-based imputation methods to test how well they perform in terms of bias and inference in hierarchical data.

Missing data can occur for a variety of reasons, ranging from respondents' refusal to answer certain questions in a survey – known as "item nonresponse" – to data loss during collection or storage processes, or intentional omission (Graham, 2012). Complete case analysis, i.e., deleting incomplete data prior to analysis, can have a significant impact on the validity and reliability of the analysis, as ignoring missing data often leads to biased results and incorrect conclusions and should therefore be avoided (van der Heijden, Donders, Stijnen, & Moons, 2006).

A prominent and widely used technique for dealing with missing data is multiple imputation (MI), which involves creating multiple reasonable values for each missing value, so that multiple complete data sets are generated (Little & Rubin, 2002). Each of the complete data sets is used separately to perform statistical analysis using standard statistical techniques. The results of the analyses are then combined using methods that account for the variability of the imputed values, resulting in more reliable estimates (Rubin, 1987). MI is particularly useful when a large amount of data is missing and the missingness mechanism is different from missing completely at random (Little & Rubin, 2002; Schafer, 2000).

Imputation models should preserve the relationships that exist in the data and account for the process that created the missing data (van Buuren & Groothuis-Oudshoorn, 2011). For hierarchical data with multiple levels, such as clustering at individual and higher-level units like classes or schools, maintaining the hierarchical

structure is challenging and often overlooked (van Buuren, 2011). Although fully conditional specification (FCS) or multiple imputation by chained equations (MICE) is a prevalent approach in the social sciences, it presents several limitations, including its considerable complexity due to the challenges of the specification of imputation model and computational intensity (van Buuren & Groothuis-Oudshoorn, 2011). Since MICE heavily relies on model specifications, it can lead to issues like overfitting and convergence errors, especially when dealing with multicollinearity and other instability problems (van Buuren & Oudshoorn, 1999). In addition, MICE is time-consuming and its use restricted in cases where, for example, data needs to be resampled or the number of variables is (too) high. Exploring alternatives to MICE, such as non-parametric tree-based methods, which are less assumption-dependent, can enhance the robustness and reliability of statistical analysis in empirical studies. Therefore, in this simulation study, we address the following research questions: Do tree-based imputation methods exhibit similar performance in terms of bias, type I error, and power compared to the standard level-2 imputation method? The specific tree-based methods are chained random forests (Stekhoven & Bühlmann, 2012; Tang & Ishwaran, 2017) and extreme gradient boosting (Deng, 2023). To accommodate the data structure, we also use dummy variables for each cluster to adapt the tree-based imputation methods by incorporating the multilevel structure of the data (Lüdtke, Robitzsch, & Grund, 2017). We examine the performance with 25 and 50 clusters to evaluate how the results depend on whether the data contain a few clusters (25) or a large number of clusters (50). Though many different methods are suggested to be used in MI, like multilevel MI with joint modeling, multilevel MI with fully conditional specifications, multilevel substantive-model-compatible MI with sequential modeling, model-based treatment with Bayesian estimation, as reviewed recently by (Grund, Lüdtke, & Robitzsch, 2024), tree-based methods have not been evaluated for multilevel data.

The remainder of the paper is organized as follows: In the next section, we briefly review multilevel data and existing (parametric) MI approaches for multilevel data structures. Next, the simulation study is detailed in Section 3, starting with a review of the imputation approaches, an explanation of the simulated missingness mechanisms and the fitted linear mixed models. We present simulation results in Section 5 also with

the help of dot plots for the relative performance, and we close in Section 5 with remarks on the current study and potential further work.

2. Multilevel data

Multilevel data structures are common in social and behavioral sciences research (Hox & Roberts, 2010). This is often seen in, e.g., educational research where students (Level 1) are nested within classes, schools, or regions (Level 2) (Hox & Roberts, 2010). The presence of higher-level variables can significantly influence the outcome variable, calling for robust analysis methods that account for the complexity introduced by these hierarchical structures (Grund et al., 2024; Steenbergen & Jones, 2002), since simply ignoring the dependencies or aggregating everything to a single level can be deceptive (Aitkin & Longford, 1986; Grund et al., 2024).

To maintain statistical integrity, it is crucial to employ appropriate imputation methods tailored for multilevel data. These methods should consider both within-cluster and between-cluster variability for a more accurate representation of the underlying data (Audigier et al., 2018; Grund et al., 2024; van Buuren, 2011). Ignoring the nested structure of data through aggregation or disaggregation to a single level is suboptimal and can result in misleading conclusions (Aitkin & Longford, 1986). To address this, multilevel modeling techniques, such as hierarchical linear models including the random intercept model as well as the random intercept and random slope model, have been developed to address the variance components properly at each level, providing a sophisticated statistical framework for analyzing hierarchical data structures (Hox, 1998; Hox & Roberts, 2010).

2.1. Existing imputation approaches for multilevel data

For the multilevel analysis to be valid, the dependency between observations (multilevel structure) should be taken into account in the imputation model. Otherwise, the estimations might be biased even when the statistical methods are appropriate (Audigier et al., 2018; Hox & Roberts, 2010). A study from Enders, Mistler, and Keller (2016) compared two imputation frameworks for multilevel data: joint modeling (JM)

and chained equation imputation (MICE with a two-level normal model `2l.norm`). The joint model turned out to be better for analysis postulating distinct within- and between-cluster relations and chained equations imputation turned out to be superior in random slope analysis (Enders et al., 2016). With JM, imputations are created according to a joint model for all variables simultaneously as draws from the fitted distribution, whereas the observations are grouped according to missing data patterns (van Buuren in Hox & Roberts, 2010, pp. 173-196).

Next to JM, another standard MI procedures for multivariate multilevel data is a *fully conditional specification* of MI (FCS). Imputations with FCS are constructed on a variable-by-variable basis. For each variable with missing parts, an imputation model is specified and imputations are generated iteratively (van Buuren, 2012).

For clustered data both of these methods are effective in the broad context of *random intercept* models, even with variables at higher levels, as simulation results by Grund, Lüdtke, and Robitzsch (2018b) indicate. However, for *random slopes* models, FCS appears to be more flexible than JM, but still has some limitations and is not all that reliable when data in an explanatory variable is missing (Grund et al., 2018b). Similar results were shown by Enders et al. (2016) for *random intercepts*. Although they also show that for *random slope* analysis, *chained equations imputation*, which is an imputation implementation under conditionally specified models, creates a substantial upgrade over JM. Additionally, Grund, Lüdtke, and Robitzsch (2018a) show that both approaches (chained equations and joint modeling) provide useful tools for dealing with missing data at Level 2 in most applications in practice (especially for balanced data).

One advantage of choosing FCS over JM is that FCS grants more flexibility in creating multilevel models by splitting a k-dimensional problem into k one-dimensional problems. I.e., for each of the variables with missing data, a regression model with a univariate outcome conditional on the other $k-1$ variables is constructed. Furthermore, it is easier to avoid logical inconsistencies in the imputed data and incorporate methods to preserve unique features in the data (van Buuren, Brand, Groothuis-Oudshoorn, & Rubin, 2006), e.g., temporal dependency can be taken into account in longitudinal data. Multiple imputation by chained equations (MICE), often used synonymously to FCS, is a specific implementation of the broader FCS framework. MICE uses the conditional

imputation approach from FCS and chains the variables by imputing iteratively one variable at a time based on the others (van Buuren & Groothuis-Oudshoorn, 2011).

3. Simulation study design

The use of tree-based methods grows in empirical research, especially with `missRanger` (Levi, Wolf, Sommer, & Howe, 2023; Sajeev, Champion, Maeder, & Gordon, 2022; Schwerter, Bleher, Doebler, & McElvany, 2023). However, it is still unclear how reliable the statistical inference is for data that has been imputed with tree-based methods. While a recent simulation study by Schwerter, Gurtskaia, Romero, Zeyer-Gliozzo, and Pauly (2023) investigated the performance of tree-based methods in longitudinal data, we focus on their performance on hierarchical data, especially when there is missingness at a higher level. How trustworthy are tree-based imputation methods for multilevel data in terms of the reliability of the statistical inference? In particular, we aim to find the strengths and limitations of tree-based imputation methods concerning type I error rates, statistical power, and coefficient bias, compared to the widely-used MICE approach, for simulated multilevel data with data missing at random, completely at random and not at random.

3.1. Multiple imputation methods

Overview of imputation methods. In this simulation study, we implemented three main imputation methods: Multiple imputation by chained equations (MICE) as our baseline since it has been shown in the literature to be superior to other imputation methods (see Enders et al., 2016; Grund et al., 2018a, 2018b). Additionally, we include more recent tree-based imputation methods: a fast implementation of random forests from the `missRanger` package (Mayer, 2021), and multiple imputation by XGBoost implemented in the `mixgb` package (Deng, 2023). For MICE the imputation methods and the prediction matrix for each variable are carefully adjusted to take into account the multilevel data structure. For Level 1 variables the Level 1 normal model (`2l.norm`) is used and for Level 2 variables the Level 2 class predictive mean matching (`2lonly.pmm`). Two factors were varied for `missRanger`: the number of predictive mean matching donors

(3 or 5)[1], and the variant (standard or an adapted implementation of `missRanger`). The adapted variant respects the multilevel structure of the data by including dummy variables for the respective cluster, while the standard implementation does not. Similarly, `mixgb` was adapted with additional dummy variables. We used 5 imputations[2] Details of each method follow.

3.1.1. Multiple imputation by chained equations

Multiple Imputation by Chained Equations (**MICE**) is a flexible and efficient imputation method that can treat missingness in a wide range of data types and analysis models. Its applications can be found in diverse research fields, including medicine, epidemiology, psychology, management, politics, and sociology (van Buuren & Groothuis-Oudshoorn, 2011). MICE derives imputations for each missing value iteratively from the other observed variables in the dataset.

The procedure described by van Buuren et al. (2006) and van Buuren and Groothuis-Oudshoorn (2011) is the following: Suppose $Y = (Y_1, ..., Y_k)$ is a random sample from the k-variate multivariate distribution $P(Y|\theta)$, which is assumed to be completely specified by a vector of unknown parameters θ (van Buuren & Groothuis-Oudshoorn, 2011, p. 6). The algorithm obtains the posterior distribution of θ by sampling iteratively from conditional distributions of the form: $P(Y_\ell|Y_{-\ell}, \theta_\ell)$, where $Y_{-\ell} = (Y_1, ..., Y_{\ell-1}, Y_{\ell+1}, ..., Y_k)$. The parameters are specific to the respective conditional densities. The t-th iteration of the algorithm contains the following successive

[1]The default value for `missRanger` is 3, and the default value in MICE is 5. Not using PMM for `missRanger` showed inflated type I errors in another context (Ramosaj, Amro, & Pauly, 2020).

[2]To choose a number of imputations a mini-simulation on randomly selected design was run with 5 and 20 imputations with MICE and `missRanger` (standard and adjusted with three donors). The results suggested very little to no differences between the number of imputations in terms of Coefficient estimation bias and H_0 rejection rates, but imputing 20 times takes approximately four times longer than performing 5 imputations (18.2145, 1.9713, 1.9248 seconds with 5 imputations and 72.7969, 8.1181, 7.8682 seconds with 20 imputations for MICE, `missRanger` and adjusted `missRanger` respectively). Consequently, the whole simulation is run with 5 imputations for each method, making the simulation considerably faster.

draws of the Gibbs sampler[3]:

$$
\begin{aligned}
\theta_1^{*(t)} &\sim P(\theta_1|Y_1^{obs}, Y_2^{(t-1)}, ..., Y_k^{(t-1)}) \\
Y_1^{*(t)} &\sim P(Y_1|Y_1^{obs}, Y_2^{(t-1)}, ..., Y_k^{(t-1)}, \theta_1^{*(t)}) \\
&\vdots \\
\theta_k^{*(t)} &\sim P(\theta_k|Y_k^{obs}, Y_1^{(t)}, ..., Y_{k-1}^{(t)}) \\
Y_k^{*(t)} &\sim P(Y_k|Y_k^{obs}, Y_1^{(t)}, ..., Y_k^{(t)}, \theta_k^{*(t)})
\end{aligned}
$$

The MICE procedure for multilevel structured data iterates between the available variables as output variables. For a *random intercept model* (M1) with k variables with missing data in all variables, a proper MI by chained equations imputes on the basis of the following univariate models:

$$
\begin{aligned}
Y_{ij}^1 &= \beta_{0(Y^1)} + \beta_{1(Y^1)}Y_{ij}^2 + \beta_{2(Y^1)}Y_{ij}^3 + ... + \beta_{k-1(Y^1)}Y_{ij}^k + \delta_{0j(Y^1)} + \varepsilon_{ij(Y^1)} \\
&\vdots \\
Y_{ij}^k &= \beta_{0(Y^k)} + \beta_{1(Y^k)}Y_{ij}^1 + \beta_{2(Y^k)}Y_{ij}^2 + ... + \beta_{k-1(Y^k)}Y_{ij}^{k-1} + \delta_{0j(Y^k)} + \varepsilon_{ij(Y^k)}
\end{aligned}
$$

Where Y_{ij}^{ℓ} indicates the ℓ-th variable, for $i = 1, ..., n_\ell$ observations in $j = 1, ..., J$ clusters. Imputing at Level 2 requires additional considerations. One possibility is to have a separate model for Level 2 variables, for example, a regression model that includes other Level 2 variables as well as cluster-level components (e.g, mean, median) of the Level 1 variables. As described by Grund et al. (2018a), imputations at Level 2 may be generated from $L_{2jk} \sim P(L_{2jk}|\tilde{L}_{1j}, L_{2j(-k)}, \theta_k)$. Where L_{2jk} are cluster j values for variable k at Level 2, $\tilde{L}_{1j}$ can be means of variables within each group at Level 1 and θ_k is the parameter of the model.

As mentioned, `mice` is an R package for chained equations imputations using a function of the same name. To impute Level 1 missing values `2l.norm`[4] is used, which uses uni-

[3]Gibbs sampler is a Markov Chain Monte Carlo (MCMC) algorithm used for drawing samples from a multivariate distribution. The algorithm iteratively draws a value for one variable from its conditional distribution, given the values of all other variables in the model (Schafer, 2000, pp. 69-70)

[4]2l indicates the data structure which is multi-level, it does not mean the calculation happens at a higher level, but it considers that there are variables at another level

variate missing data imputation with a two-level normal model. For Level 2 variables, a level-2 class predictive mean matching (`2lonly.pmm`) is applied. Although Enders et al. (2016) used `2lonely.norm`, we use `2lonly.pmm` because `2lonely.norm` follows the normality assumption, which does not hold in our case because we transformed some variables to aggregate them at Level 2. `2lonly.pmm` is a semi-parametric method, which is why it works in a wider range of cases.

3.1.2. Chained random forest imputation

Random forests (RF) build and combine multiple decision trees to improve upon a single decision tree's stability and precision. Each tree uses a random subset sampled from the data independently and with the same distribution for all trees in the forest, increasing generalizability and reducing overfitting (Breiman, 2001). After a large number of trees are generated, the final prediction is chosen, either with a majority vote (for categorical variables) or as the average (for continuous variables) of the predictions of the individual trees. Since RF can handle mixed types of data, is capable to address interactions and nonlinearity, and does not overfit (because of the Law of Large Numbers, Breiman, 2001), RF is an attractive tool for imputing missingness (Stekhoven & Bühlmann, 2012; Tang & Ishwaran, 2017). `missForest` is a non-parametric MI method introduced by Stekhoven and Bühlmann (2012), which converts the problem of missingness into a prediction problem. In a first step, starting values for all missing values are generated. Iteratively, a random forest for each variable to be imputed is fitted, based on current predictions of missing values using currently fitted random forest for the other variables (Tang & Ishwaran, 2017). This means that for k variables, k forests are to be fitted in each iteration, which could be challenging and is slow for certain multivariate problems.

The package *ranger*, a memory-efficient, parallelized and hence fast implementation of random forests for high dimensional data (Wright & Ziegler, 2017) can process large data sets. For splitting, the feature values are either sorted beforehand and accessed by their index, or the raw values are retrieved and sorted while splitting (Wright & Ziegler, 2017). In accordance with Breiman (2003)'s design of how RF works, the algorithm randomly selects a subset of data (of the same size) with replacement, i.e., a

bootstrap sample. K, the number of variables to be selected is specified, and for each bootstrapped sample K variables are randomly chosen (therefore, the trees are random and diverse).

Mayer (2021) used the *ranger* implementation of RF to develop a fast missing value imputation method by chained random forests based on *missForest*. `missRanger` allows for fast multivariate imputation. Additionally and contrary to `missForest`, it also offers the option of *predictive mean matching* (Mayer, 2021). The basic steps for MI with `missRanger` are (i) splitting the data into complete and missing parts. (ii) For each variable with missing values, RF is fitted based on other variables, and (iii) the fit is used to predict missing values. Then, (iv) a set of reasonable imputed values is created based on the trained RF models (Breiman, 2003; Mayer, 2021; Stekhoven & Bühlmann, 2012). Optionally for PMM, (v) for each predicted value in the test set, one of the closest k values in the trained set is randomly chosen and its observed value is returned. The number of donors k is an adjustable tuning parameter. For multiple imputations, this process is repeated multiple times (5 times in our simulation). In the decision tree construction, the *Extremely randomized trees* (ExtraTrees) algorithm is used for splitting, creating diverse set of trees (Geurts, Ernst, & Wehenkel, 2006). The procedure takes a random subset of the variables that will be considered at each split (different for each tree). For each variable, several possible thresholds are randomly generated. The candidates are then evaluated and the threshold that ensues the best split is chosen. The node is split on the threshold and the procedure is continued recursively until the criteria for stopping the algorithm are met.

3.1.3. MI through extreme gradient boosting (XGBoost)

Extreme Gradient Boosting (XGBoost) is a machine learning algorithm that belongs to the family of gradient boosting methods (Chen & Guestrin, 2016). Instead of simply taking the mean of multiple decision trees, XGBoost uses gradient boosting to combine multiple regression trees. It also employs regularization and shrinkage techniques. Typically, the depth of the regression trees used in sequential boosting is not very deep. One advantage of XGBoost over Random Forest is that XGBoost selects only one of a set of highly correlated features, because in sequential boosting, features are added to

the model incrementally. If a feature is selected early, other highly correlated features can only improve the prediction if they provide new information.

XGBoost uses gradient descent optimization techniques to iteratively minimize the loss function, making it highly efficient and effective in finding the optimal solution. L1 and L2 regularization prevent overfitting and improve generalization. Tree pruning removes unnecessary branches and reduces model complexity, further improving XGBoost's predictive performance. Lastly, XGBoost can take advantage of parallel processing capabilities, making it suitable for large datasets and reducing training time. Overall, XGBoost is known for its ability to handle complex datasets, handle missing values, and provide accurate predictions. Its popularity is evident in various domains, including Kaggle competitions and real-world applications (Chen & Guestrin, 2016).

The `mixgb R` package uses XGBoost to implement missing value imputation in a scalable and efficient manner (Deng & Lumley, online first). It addresses the challenge of missing data in large datasets with complex structures. `mixgb` imputes missing values in the order of variables with fewer missing values. This strategic approach aims to prioritize variables with more available information during the imputation process. Initial values for imputation are filled with random values drawn from the observed data. This helps kickstart the imputation process (Suh & Song, 2023). `mixgb` provides a versatile approach to missing data imputation, leveraging XGBoost, bootstrapping, and PMM to enhance imputation accuracy, especially for continuous data. Unlike some other imputation frameworks, the default imputation in `mixgb` is non-iterative, but the package allows users to set the number of iterations for imputation (Deng & Lumley, online first), for this simulation we used 5 iterations.

3.1.4. Adapting tree-based imputation methods with cluster membership dummy variable

By default, both, `missRanger` and `mixgb`, are unaware of the multilevel data structure. We study variants of the two procedures that add dummy variables for cluster membership (Lüdtke et al., 2017). If there are J clusters, J dummy variables are added, and the jth variable is equal to 1 if the data in row i is from cluster j and 0 otherwise. For the tree methods, in contrast to the regression methods as in Lüdtke et al. (2017),

no reference group is necessary. Adding the dummy variables is possible in all standard software packages. Potentially, cluster-level effects can now influence the imputations, since the models can generate cluster-specific predictions.

If regression trees have high enough tree depth, the flexibility gained by the dummy variables is potentially higher compared to a procedure that would include some kind of random intercepts, since some interactions with the dummy variables are possible, though not all kinds of interactions (Wright, Ziegler, & König, 2016). Potentially, more dummy variables are added than there are variables originally, so the dummy variable procedure relies on imputation methods that have some form of regularization. Tree-based ensemble methods like RF and XGBoost do have this property by design (Breiman, 2001; Chen & Guestrin, 2016).

3.2. Simulated data

For the simulation setup, four general factors are varied, resulting in a total of 16 simulation designs. The varying factors are: **number of clusters, data generation model, missing rate, and missing mechanism**. There are two numbers of clusters: 25 and 50. This results in two cluster sizes of 40 and 20 for a balanced design with a sample size of $\boldsymbol{N} = 1000$. This variation in the number of clusters allows us to examine the effect of cluster size on the performance of the imputation methods. As for the data generation process, we first simulate six covariates using the `fungible` package in `R` (Waller, 2022). The `monte` function generates clustered data with predefined characteristics. For this simulation study, intra-cluster correlations and indicator validities (cluster separations for each variable) are randomly constructed. Four variables are Level 1 (individual level) variables and two variables are aggregated to Level 2. All covariates are continuous numerical variables. Finally, two data generation models are considered for the outcome variable: *random intercept* and *random intercept and random slope* (hereafter denoted as *random slope* model). Based on the model, the output variable is entered with

$$
\begin{aligned}
Y_{ij}^{o} = & \, 0.3 \\
& + \delta_{0j} + 0.5 \times Y_{ij1} + 1 \times Y_{ij2} + 0 \times Y_{ij3} + 1.3 \times Y_{ij4} \\
& + 1.4 \times L_{ij1} + 0.7 \times L_{ij2} + \epsilon_{ij}
\end{aligned}
$$

for random intercept model and

$$
\begin{aligned}
Y_{ij}^{o} = & \, 0.3 \\
& + \delta_{0j} + (0.5 + \delta_{1j}) \times Y_{ij1} + (1 + \delta_{2j}) \times Y_{ij2} + 0 \times Y_{ij3} + 1.3 \times Y_{ij4} \\
& + 1.4 \times L_{ij1} + 0.7 \times L_{ij2} + \epsilon_{ij}
\end{aligned}
$$

for random slope model, where $\delta_{0j}, \delta_{1j}, \delta_{2j}, \epsilon_{ij} \sim \mathcal{N}(0, 1)$ are randomly generated.

3.2.1. Missingness mechanism

Missingness in the generated data is induced at two levels: moderately low 10% and relatively high 50%. This range of missingness levels reflects real-world scenarios. Two missingness mechanisms (MAR, MCAR) are considered. The introduction of missing data is based on the selected missingness mechanism and the specified missing rate. For MCAR, a simple function is implemented in `R` that randomly sets each data point to **NA** with probability equal to the missingness level for each variable. To introduce missingness according to the MAR mechanism, the algorithm described by Thurow, Dumpert, Ramosaj, and Pauly (2021) is used, with slight adjustments for purely numerical data. The algorithm operates as follows: starting with one variable it generates missing values under MCAR with the overall missing rate. The missingness in all other variables depends only on the remaining (observed) data in the first variable to exhibit MAR missingness. After introducing missingness in the first variable, it is converted to a categorical variable by grouping values into intervals. For each subsequent variable i.i.d. uniformly $(0, 1)$ distributed random numbers are generated, one for each unique value of the first variable, and treated as probabilities. The probabilities are assigned

to the values of the selected variable. Then the probability of obtaining a missing value in that variable is calculated according to the assigned probability and the absolute frequencies of the distinct values in the converted variable. Lastly, indices are randomly selected based on the computed probabilities and the corresponding values are set to missing for each variable. Thurow et al. (2021) argue that this leads to the desired overall missingness rate.

The resulting simulation has $2 \times 2 \times 2 \times 2 = 16$ conditions. Each combination of the above factors is replicated 1,000 times.

3.3. Evaluation using linear mixed models

One way to analyze multilevel (clustered observations, repeated measures, longitudinal, multivariate) data without ignoring existing correlations is to use linear mixed effects models (**LMM**, Grund et al., 2024). A major advantage of **LMM** is that it allows the estimation of individual or group specific effects while taking into account the nested structure or the correlation caused by the multilevel structure. Ignoring the multilevel data structure can lead to potentially spurious standard errors and excessive type I error rates (Grund et al., 2024; Steenbergen & Jones, 2002). Because a critical assumption for classical linear models is the independence of the observations collected. This is not the case for multilevel hierarchical structures, where Level 1 observations are not independent; for example, students from the same school may have similarities, even more so within a given class, so classical linear models may not be appropriate for analyzing such data. However, they are a good starting point and can be extended to more general models for multilevel data that allow relaxing the assumptions of independence and variance homogeneity. In particular, *LMM* allows the inclusion of the correlation of observations contained in a data set (Gałecki & Burzykowski, 2013). Mixed effects models, as the name implies, include both *fixed* and *random* effects. *Fixed effects* are constant parameters that do not vary between clusters. *Random effects*, on the other hand, can be considered as random variables and are not observable (Pinheiro & Bates, 2004).

For a single level of grouping, the classical LMM, at a given level i of the grouping factor is defined by (Fahrmeir, Kneib, Lang, & Marx, 2013; Gałecki & Burzykowski,

2013; Pinheiro & Bates, 2004) as follows:

$$\boldsymbol{y}_i = \boldsymbol{X}_i\boldsymbol{\beta} + \boldsymbol{Z}_i\boldsymbol{b}_i + \boldsymbol{\varepsilon_i} \tag{1}$$

where n_i is the ith group size, $\boldsymbol{y}_i$ is the n_i-dimensional response vector, $\boldsymbol{X}_i$ is the $n_i \times p$ fixed effects design matrix, $\boldsymbol{\beta}$ is the p-dimensional unknown parameter of the fixed effects, $\boldsymbol{\varepsilon_i}$ is a vector of residual errors for group i (within-group error), and $\boldsymbol{Z}_i$ and $\boldsymbol{b}_i$ are the random effects regressor matrix and the corresponding random effects vector.

The q covariates of the matrix $\boldsymbol{Z}_i$ are known, and their corresponding effects $\boldsymbol{b}_i$ are unobservable. Both $\boldsymbol{b}_i$ and $\boldsymbol{\varepsilon}_i$ follow a multivariate normal distribution $\boldsymbol{b}_i \sim \mathcal{N}_q(\boldsymbol{0}, \boldsymbol{\mathcal{D}})$, $\boldsymbol{\varepsilon}_i \sim \mathcal{N}_{n_i}(\boldsymbol{0}, \boldsymbol{\mathcal{R}_i})$. Furthermore, the residual errors are independent of the random effects. It is also assumed that for different groups ($i \neq i'$) $\boldsymbol{b}_i$ is independent of $\boldsymbol{\varepsilon}_{i'}$. For an unknown scale parameter σ^2, the positive definite matrices $\boldsymbol{\mathcal{D}}$ and $\boldsymbol{\mathcal{R}}_i$ can be defined as follows $\boldsymbol{\mathcal{D}} = \boldsymbol{D}(\boldsymbol{\theta}_D)$ for the variance-covariance matrix of the random effects $\boldsymbol{b}_i$, with a vector of parameters $\boldsymbol{\theta}_D$. There are no other restrictions on $\boldsymbol{\mathcal{D}}$ (Gałecki & Burzykowski, 2013). Finally, $\boldsymbol{\mathcal{R}_i} = \sigma^2\boldsymbol{R}_i$, which is not identifiable in its general form. For i.i.d. errors, $\boldsymbol{R}_i$ simplifies to $\boldsymbol{I}_{n_i}$ (Fahrmeir et al., 2013).
LMM, as described above, implies that marginally

$$\boldsymbol{y}_i \sim \mathcal{N}_{n_i}(\boldsymbol{X_i\beta}, \boldsymbol{Z_i D}(\boldsymbol{\theta}_D)\boldsymbol{Z}_i' + \boldsymbol{\sigma}^2\boldsymbol{I}_{n_i}) \tag{2}$$

Equation 1 can be specified for all data in a compact by defining vectors (Fahrmeir et al., 2013; Gałecki & Burzykowski, 2013): Let $\boldsymbol{Y} = \begin{pmatrix} y_1' & \cdots & y_m' \end{pmatrix}$, $\boldsymbol{b} = \begin{pmatrix} b_1' & \cdots & b_m' \end{pmatrix}$ and $\boldsymbol{\varepsilon} = \begin{pmatrix} \varepsilon_1' & \cdots & \varepsilon_m' \end{pmatrix}$ for all groups $i = 1, ..., m$, and set the design matrices to

$$\boldsymbol{X} = \begin{pmatrix} X_1 \\ \vdots \\ X_m \end{pmatrix} \text{ and } \boldsymbol{Z} = \begin{pmatrix} Z_1 & 0 & \cdots & 0 \\ 0 & Z_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & Z_m \end{pmatrix}$$

Then

$$\boldsymbol{Y} = \boldsymbol{X\beta} + \boldsymbol{Zb} + \varepsilon \quad \text{with} \quad \begin{pmatrix} \boldsymbol{b} \\ \boldsymbol{\varepsilon} \end{pmatrix} \sim \mathcal{N}\left(\begin{pmatrix} 0 \\ 0 \end{pmatrix}, \begin{pmatrix} \boldsymbol{D(\theta_D)} & 0 \\ 0 & \boldsymbol{\sigma^2 I} \end{pmatrix} \right) \tag{3}$$

Depending on the research question, Equation 3 can also be defined more specifically. If the slope is the same for all Level 1 individuals (e.g., students, patients) in all groups (*constant slope* $\boldsymbol{\beta_1}$), but the intercept ($\boldsymbol{\beta_{0j}}$) is group specific and contains random effects, then the resulting model will be a ***random intercept model***. On the other hand, if the intercept is constant for all groups ($\boldsymbol{\beta_0}$), but there are group-specific slopes ($\boldsymbol{\beta_{1j}}$), then it is a ***random slope model***. If both the intercept and slope are random and vary between groups ($\boldsymbol{\beta_{0j}}$ and $\boldsymbol{\beta_{1j}}$ for group j), then it is a ***random intercept and random slope model***.

4. Simulation Study Results

In this simulation study, various imputation methods were evaluated and compared to address missing data in multilevel designs. The primary objective was to assess the performance of these methods in terms of decision making, accuracy and overall robustness across different settings. The simulated data was designed to mimic real-world scenarios where multilevel structures are quite common with missingness patterns that are commonly encountered in diverse research settings.

4.1. Random intercept results

4.1.1. Rejection rates

Figure 1 displays the average H_0 ($\beta = 0$) rejection rates over replications for random intercept models by missingness mechanism. Notably, as Figure 1a shows, among the different imputation methods evaluated, only MICE consistently achieved a type I error rate below 5% in the presence of MCAR missingness when the true underlying coefficient was indeed zero, `missRanger` only for 10% missingness. `mixgb` had the highest type I errors for 10% missingness but was consistently closest to the com-

plete data results. For 50% missingness, dummy-adjusted `mixgb` and even standard `mixgb` performed at least as well (for 50 clusters) or better (for 25 clusters) than MICE for non-zero Level 1 coefficients. For Level 2 coefficients, standard `mixgb` outperforms dummy-adjusted `mixgb` and is only beat by MICE. All methods, except the standard implementations of `missRanger`, perform better for 50 clusters than for 25.

The results were similar for MAR missingness with 10% missing data (Figure 1b). For the zero coefficients again only MICE was consistently below 5% rejection rate, `missRanger` only for 10% missingness and `mixgb` was again close to the generated data results. For 10% missingness, `missRanger` had the desired 5% rejection rate, but its performance deteriorated at 50%. For all non-zero ($\beta \neq 0$) Level 1 coefficients except the intercept and 10% missingness, all methods had high rejection rates, but power decreased with increasing missingness. The ranger-based methods had no power for testing the non-zero intercept, but MICE and mixgb had some power, albeit non at 50% missingness rate and 25 clusters. Dummy-adjusted variants of imputation methods always improved the statistical power over standard Level 1 implementations, especially for `mixgb`, which performed almost as well and sometimes even better than MICE.

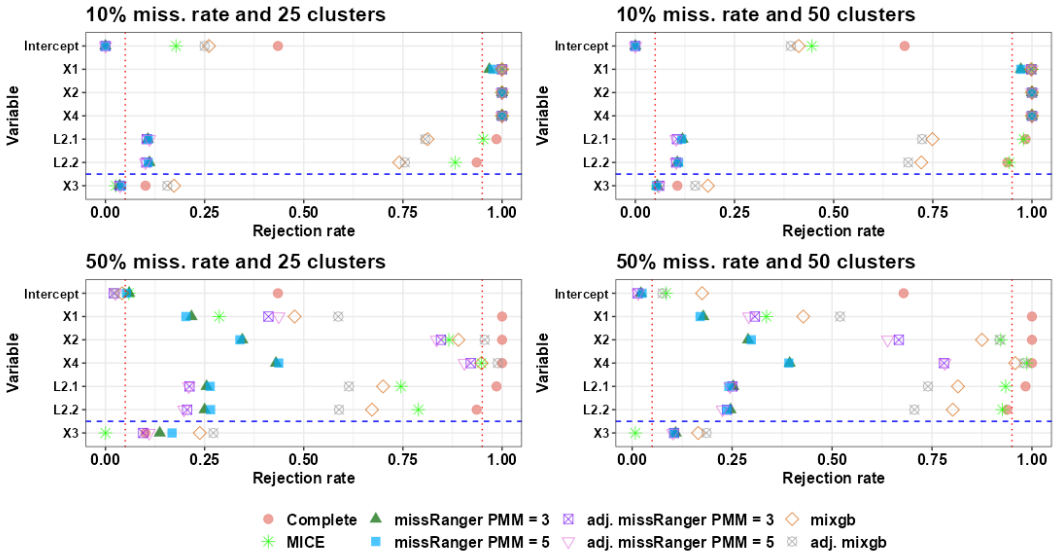
4.1.2. Coefficient bias

Figure 2 presents the coefficient bias of each method for random intercept models under MCAR and MAR. Under MCAR missingness (Figure 2a), both `mixgb` methods have the lowest estimation bias for Level 1 variables with 10% missingness. For the true zero coefficient (X3) all the methods have low bias. For Level 2 variables, all `missRanger` implementations have high coefficient estimation bias which aligns with the worst test decisions (least power). As the missingness rate increases, so does the bias, especially for MICE. At 50% missingness, MICE has the highest estimation errors for one of the Level 1 variables, except for the true zero coefficient, where MICE still has the lowest bias.

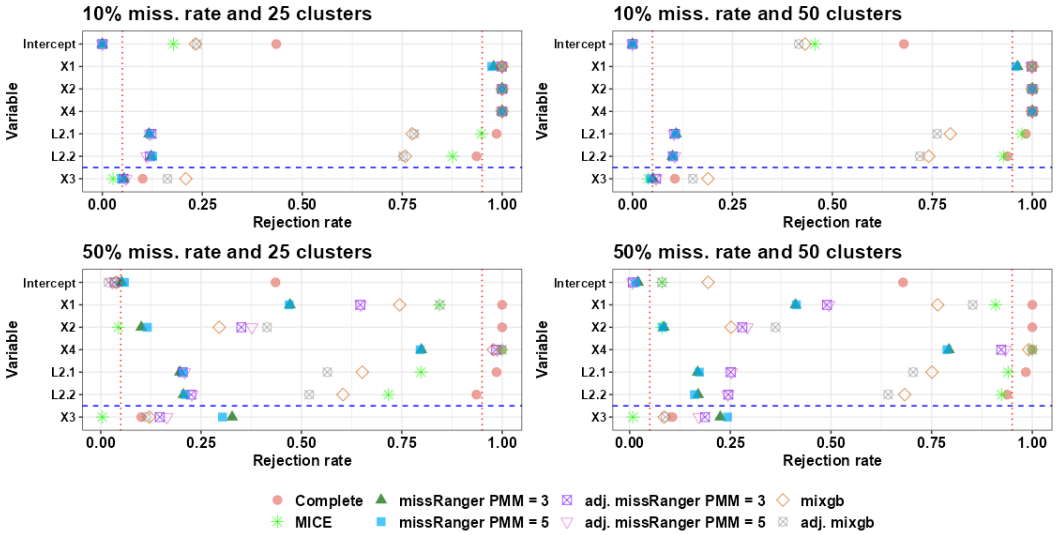
The situation is a bit different with MAR mechanism (Figure 2b). For 10% missingness, MICE has the highest bias for Level 1 and one of the Level 2 variables and `mixgb` has the lowest. With increasing missingness, standard `missRanger` sometimes even reduces bias (variables X1, X4) to almost zero or increases the bias the most (X2).

Figure 1.: Rejection rates for random intercept model

(a) MCAR



(b) MAR

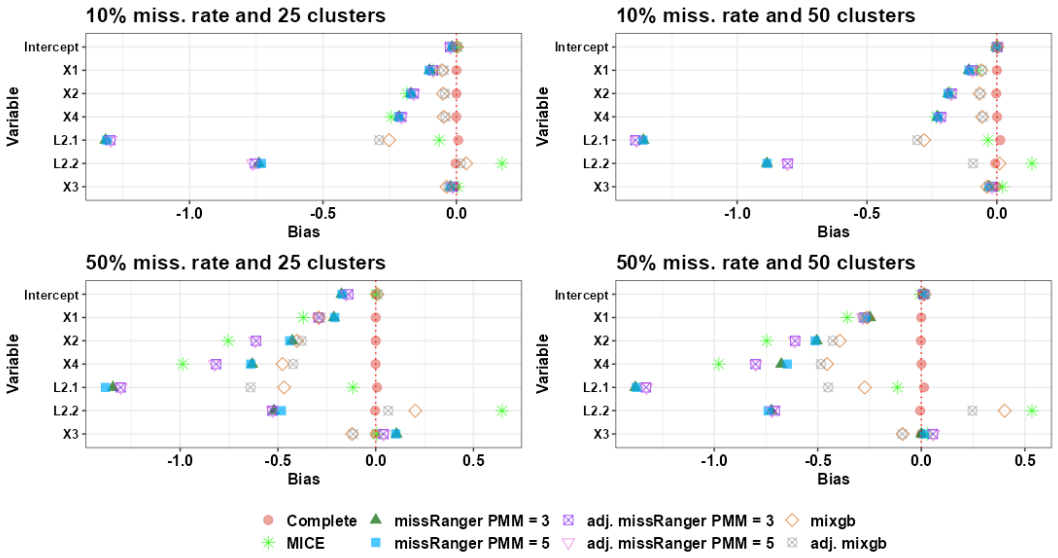


Note: Variable X3 has a true-zero coefficient.
Rejection rates for random intercept designs with 10% (first rows) and 50% (second rows) missingness and 25 (first columns) and 50 (second columns) clusters

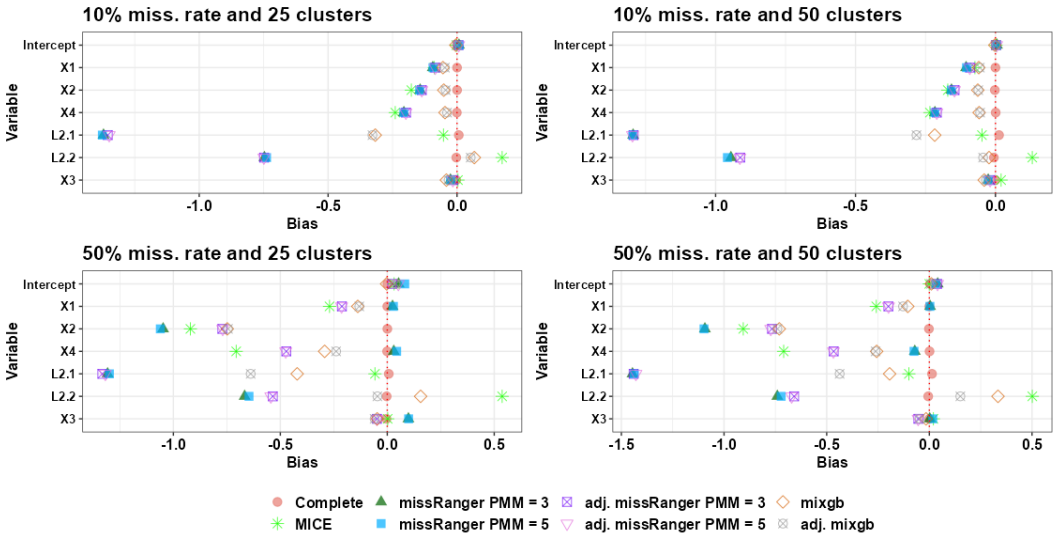
Among the imputation methods (for MAR and MCAR), the two standard `missRanger` variants are the least affected by the increasing missingness rate, however, especially the bias for the Level 2 variables and some Level 1 variables is substantial.

Figure 2.: Coefficient Bias for random intercept model

(a) MCAR



(b) MAR



Note: Variable X3 has a true-zero coefficient.
Coefficient estimation bias for random intercept designs with 10% (first rows) and 50% (second rows) missingness and 25 (first columns) and 50 (second columns) clusters

4.2. Random slope results

4.2.1. Rejections rates

For random intercept and random slope models, Figure 3 displays the average rejection rates over replications grouped by the missingness mechanism. Under MCAR (Figure 3a), both `missRanger` variants and MICE have the desired type I error (5%) for the true zero variable with 10% missingness. Both `mixgb` implementations are closer to the rejection rate of the simulated data, which is around 10%. For X1, one the coefficients with random slopes, none of the methods reach the rejection rate of the complete data, although all but the unadjusted `missRanger`-methods are close. All methods have higher rejection rates with 50 clusters than with 25 clusters. Dummy-adjusted `missRanger` is better than standard `missRanger`, especially for 50 clusters. For X2, also a variable with a random slope, all methods have a rejection rate of 100%. For Level 2 variables, the `missRanger` variants have the lowest power and MICE has the highest. `mixgb` has less power than MICE but much more power than `missRanger`.

As the missingness rate increases, all methods perform considerably worse. For the true zero coefficient, only MICE retains a type I error below 5%. Standard `missRanger` and dummy-adjusted `mixgb` lose more power for 25 clusters than for 50. For X1 all the rejection rates drop below 50%, especially MICE loses the most power. Dummy-adjusted `mixgb` has the highest rejection rate. Standard `missRanger` loses a lot of power with increasing missingness for X2 (rejection rate below 50%). Dummy-adjusted `missRanger` shows better performance than MICE with 25 cluster and MICE has higher rejection rate with 50 clusters. Dummy-adjusted `mixgb` has the highest power in both cases. For Level 2, the performance of `missRanger` improves, especially the standard implementations. The standard `mixgb` performs better than the adjusted one and even increases the power with 50 clusters. MICE still has the highest rejection rates.

The methods perform similarly for 10% under MAR (Figure 3b) as well. With increasing missingness rate, all methods lose power. MICE remains as the only method with a type I error under 5% for the true zero coefficient, whereas the standard implementations of `missRanger` has the highest rejection rate especially with 50 clusters. For X2 MICE loses the most power and is too conservative. Dummy-adjusted `mixgb`

performs best among the imputation methods for coefficients with random slopes, especially with 50 clusters. For X4 `mixgb` and MICE remain over 95% rejection rate. For Level 2 variables, MICE once again has the highest rejection rate, followed by `mixgb` and dummy-adjusted `mixgb`. While the standard implementations of `missRanger` still have low power, the rejection rates for dummy-adjusted `missRanger` even increases for a higher rate of missingness, especially for 50 clusters, where the power almost doubles with 50% missing rate.

4.2.2. Coefficient bias

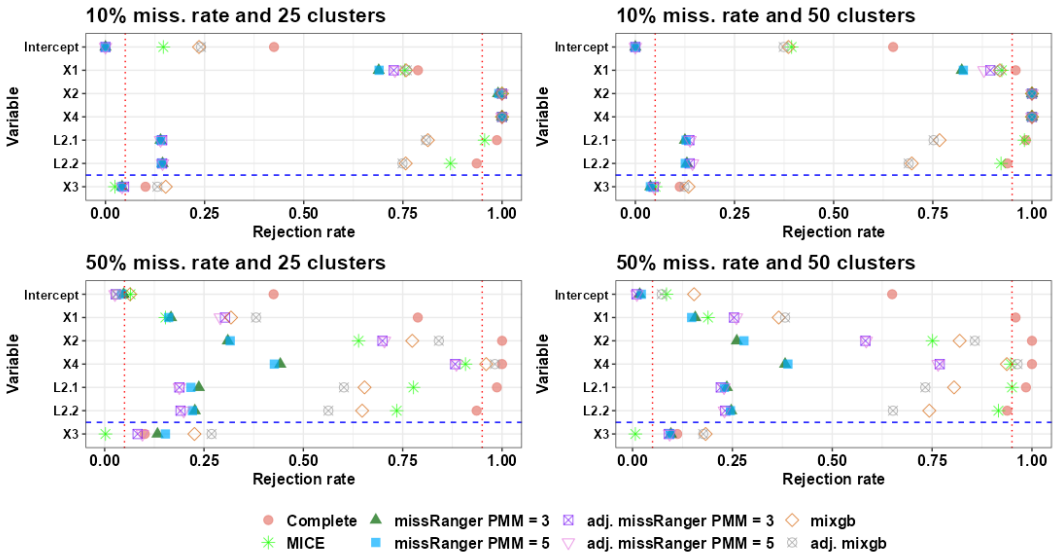
Under MCAR (Figure 4a) and 10% missingness, both `mixgb` approaches have the lowest bias for Level 1 variables ($\beta \neq 0$). MICE has the highest bias with 25 clusters. For Level 2 variables `missRanger` has considerably higher bias than other methods while MICE or standard `mixgb` have the lowest. For the true zero coefficient both `mixgb` have slightly higher bias than the others.

The bias increases with increasing missingness. At Level 1, the bias for MICE increases the most and becomes the highest for both 25 and 50 clusters. Standard `missRanger` remains more stable and has lower bias rate than dummy-adjusted `missRanger` and standard `mixgb` has the lowest bias for 50 clusters. At Level 2, `missRanger` increases its bias the least. MICE remains stable for one coefficient but the error rate more than triples for the other. For these coefficients, dummy-adjusted `mixgb` has the lowest bias. For the true zero coefficient, both `mixgb` once again have the highest bias.

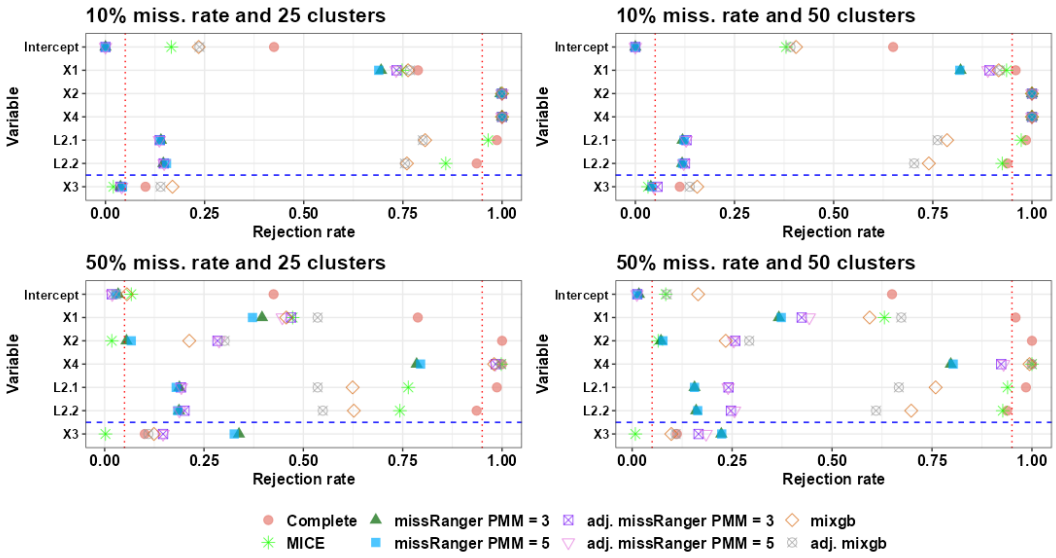
Figure 4b for MAR mechanism looks somewhat similar. With 10% missingness, `mixgb` has the lowest bias at Level 1 ($\beta \neq 0$) and MICE has the highest. At Level 2, `missRanger` has the highest bias and either MICE or standard `mixgb` has the lowest. The positions change slightly at 50% missingness, `missRanger` has either the highest (X2) or lowest (X1 and X4) bias, while MICE has the highest for X1 and X4. At Level 2 the situation is similar to MCAR, except that `missRanger` actually reduces the bias with increasing missingness, especially dummy-adjusted variations.

Figure 3.: Rejection rates for random slope models

(a) MCAR



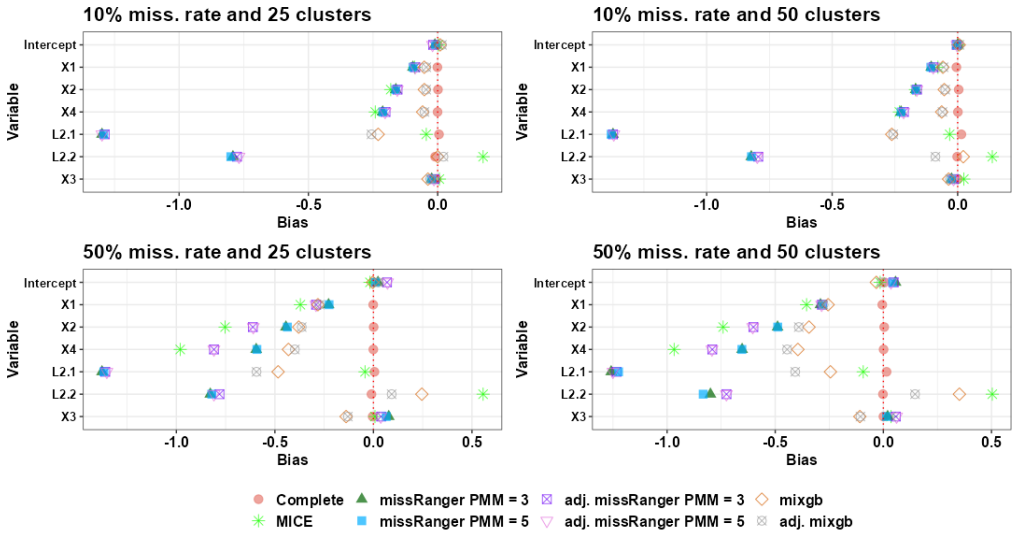
(b) MAR



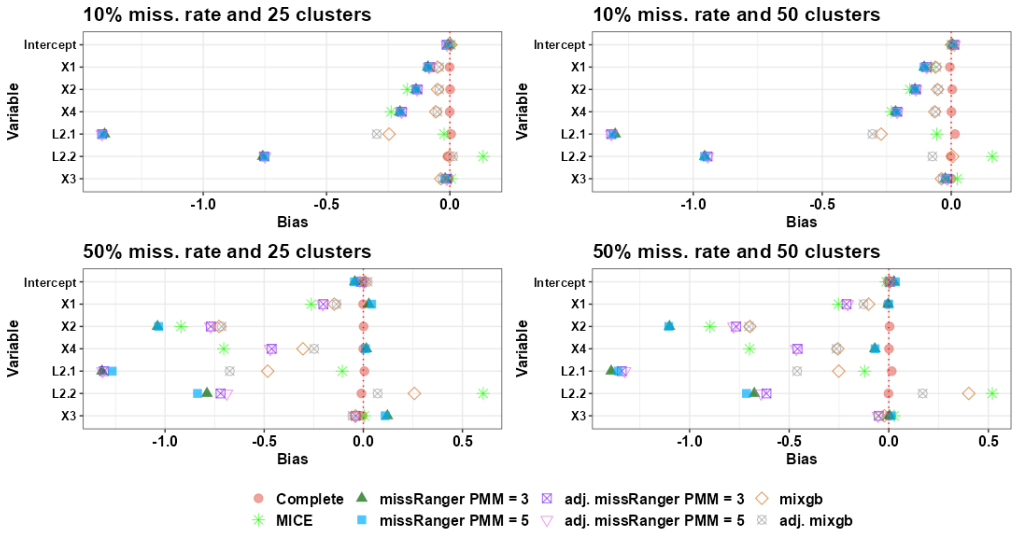
Note: Rejection Rates for random slope designs with 10% (first rows) and 50% (second rows) missingness and 25 (first columns) and 50 (second columns) clusters

Figure 4.: Coefficient Bias for random slope models

(a) MCAR



(b) MAR



Note: Coefficient estimation bias for random slope designs with 10% (first rows) and 50% (second rows) missingness and 25 (first columns) and 50 (second columns) clusters

5. Discussion

This paper critically evaluates the performance of novel tree-based imputation methods for handling missing data in hierarchical data structures for Level 1 and Level 2 variables with two different data generation processes (random intercept and random slope), two missingness rates (10% and 50%), and two missingness mechanisms (MCAR and MAR). Through a comprehensive simulation study, we contrast these novel techniques with the more conventional multiple imputation MICE, focusing in particular on bias and inference. Our results indicate that MICE is characterized by consistent accuracy in rejection rates when using the `2l.norm` method for Level 1 variables and `2lonly.pmm` for Level 2 variables. This consistency underscores MICE's robustness to hierarchical data and closely approximates true rejection rates (i.e., type I error and power). Our results are consistent with the findings in Grund et al. (2018a), demonstrating the practical usefulness of MICE while dealing with missingness in multilevel structures, especially in the case of missingness at Level 2. Similar to Enders et al. (2016), we also found that MICE was superior, not just for random slope but for random intercept models as well, when the imputation model was specified according to data generation process.

In addition, MICE consistently outperforms other methods across missingness mechanisms and rates in cases involving true zero Level 1 coefficients. This aspect underscores MICE's reliability in accurately identifying non-significant variables and reinforces its position as a versatile tool in statistical analysis. As in multi-level multiple imputation dependencies have to be taken into account (Audigier et al., 2018), tree-based method might miss this compared to our MICE framework. For example, for the random intercept model, in scenarios with 10% missingness on Level 2 variables, both `mixgb` variants and MICE have both good type I error and power over different settings (25 vs. 50 clusters and random intercept vs. random slope models), but at 50% missingness, the rejection rates worsen for `mixgb`, while they remain mostly robust for MICE. Such variability in performance based on missingness rate and imputation method indicates the context-dependent effectiveness of each method. However, tree-based methods, especially Mixed Gradient Boosting (`mixgb`), exhibit lower biases,

suggesting their potential in scenarios where bias reduction is a priority. This finding is particularly noteworthy as it points to the evolving capabilities of non-traditional methods in dealing with missing data.

We included the 25 and 50 cluster scenarios because when data are clustered, for asymptotically valid standard errors, not only the number of observations but also the number of clusters must go to infinity (i.e., be greater than 30, Cameron & Miller, 2015). Our results confirm that with 50 clusters, the differences between 10% missingness and 50% missingness are smaller than with 25 clusters, where the differences between 10% missingness and 50% missingness get worse. Thus, having more clusters helps to get more reliable rejection rates. More specifically, for the random intercept model, fewer clusters generally showed lower performance, especially notable for `mixgb` and MICE, while `missRanger` showed minimal difference. The trend in type I error rates under the random intercept model did not show a clear pattern. However, in the random slope model, power tended to be lower with fewer clusters, and there were slightly higher type I errors in scenarios with fewer clusters. In terms of bias, the random intercept model with 10% missing data and 25 clusters in the MCAR condition showed the least bias. Beyond this specific case, the difference in bias between the two cluster sizes remained marginal. A similar trend was observed for the random slope model, where the variation in bias between different cluster sizes was negligible, indicating a relative consistency in bias across cluster sizes.

The results do not necessarily show that adjusted tree-based methods (which also include dummies for the Level 2 cluster) outperform standard tree-based methods. For example, for MCAR and MAR, standard `mixgb` had higher power than adjusted `mixgb` for Level 2 coefficients with 50% missingness for both random intercept and random slope, while adjusted `mixgb` sometimes had higher power for Level 1 coefficients when not similar. The differences for `missRanger` were generally smaller than for `mixgb`. Thus, whether to use standard or adjusted tree-based imputation methods depends on whether the main variable of interest is at Level 1 or Level 2. For example, adjusted `mixgb` performed very similarly for Level 1 coefficients with 10% and 50% missingness. In our simulation example, none of the `missRanger` variants outperformed MICE or `mixgb`, although it was reliable for Level 1 coefficients. This is in contrast to (Schw-

erter, Gurtskaia, et al., 2023), a simulation with longitudinal data without hierarchical structure, where MICE `missRanger` (and Random Forest, not tested here) performed better than `mixgb` (and MICE PMM, not tested here).

The general picture is very similar for the random slope data generation process. E.g., at Level 2, both, standard `missRanger` PMM = 3 and PMM = 5, have higher rejection rates for MCAR with high missingness. For random slopes model under MAR adjusted `missRanger`, the power for Level 2 variables with high missingness improves, especially with 50 clusters the power almost doubles. Although MICE generally has the best rejection rates, it shifts for the random slope model when looking at Level 1 coefficients with 50% missingness. In this particular case, `mixgb` outperforms MICE in most cases, as `mixgb` is less affected by the increase in missingness. In addition, MICE suffers more from a few clusters moving from low to high missingness than for 50 clusters. Finally, `missRanger` performs better under MAR than MCAR for Level 2 coefficients.

It is unclear how the differences in rejection rates relate to the differences in bias. In general, MICE has better rejection rates and `mixgb` has lower bias, which would disqualify the hypothesis that the higher the bias, the worse the rejection rate. However, within models, moving from low to high missingness, such a pattern can be seen: The higher bias under MCAR for high missingness is associated with a worse rejection rate Level 1 coefficients for MICE. Similarly, under MAR, all `missRanger` variants, especially standard `missRanger` become more biased for high missingness, and power decreases and type I error increases.

Based on these mixed results, recommendations for which method to use are very case-specific. If researchers can be confident that the true data generation process follows a random intercept or random slope model, MICE is likely to provide more reliable rejection rates than tree-based methods, especially with a high number of clusters. In cases where rejection rates are not important, but bias is, `mixgb` should be used to impute missing data. If MICE cannot be used (due to too many variables, uncertain data generation model, high collinearity between variables, too time-consuming calculations or similar), adjusted tree-based methods that include dummies for the clusters should be used if Level I coefficients are of interest, while standard tree-based methods should

be used for Level 2 coefficients. In most cases, `mixgb` seems to outperform `missRanger` and should therefore be the first imputation alternative for multilevel data. While the differences between (standard and adjusted) `missRanger` PMM = 3 and PMM = 5 are small, in cases where there are differences, PMM = 5 outperforms PMM = 3. Thus, when using `missRanger`, it is advisable to increase the number of donors to 5.

There is no "one size fits all" solution: For the data generation process following both random intercept and random slope models, there was one variable (X2) where all methods become significantly worse with higher missingness under MAR. Especially MICE and standard `missRanger` with PMM = 3.

Of all the imputation methods evaluated, MICE proved to be the slowest (see Figures A1 and A2 in the appendix). In general, MICE needs 8-10 times longer for imputation than `missRanger`. Therefore, the efficiency of MICE should be considered in light of its computational requirements. Furthermore, while including the cluster dummies only increased the time for `missRanger` with PMM = 5 (up to the same level as `mixgb`), adjusting `mixgb` with dummy variables made it significantly slower, almost always doubling the total imputation time.

5.1. Limitations, strengths and outlook

Although the study is comprehensive, it is not without limitations. The simplicity of the data structure used in our simulations, characterized by a low number of variables, contrasts with the often complex and variable-rich data sets encountered in real-world scenarios. As a result, the generalizability of our results to more complicated data sets is uncertain. The rather simple dataset could also be a reason why MICE has more reliable rejection rates than the tree-based methods. One advantage of tree-based methods is handling large numbers of variables, that cannot shine here in our relatively low-dimensional data typical for applications of LMMs. In addition, among all imputation methods, MICE is the one that best fits the true underlying data generation process, but this process may not be known in empirical data. Future research investigating the effect of increasing the number of variables on both rejection rate and bias could provide a more complete understanding of the performance of the methods in more complex data settings. Especially, simulating data beyond a linear (mixed) model could prove

interesting, e.g., a semi- or non-parametric data generating model that is nevertheless plausible enough for an LMM-analysis.

The strength of the current study, however, lies in its novelty. We introduce and evaluate the combination of the latest tree-based imputation methods and a simple multilevel adjustment via dummy variables against established techniques such as MICE. This approach provides new insights into the evolving landscape of data imputation methods, especially in the context of hierarchical data structures.

Another interesting area to explore would be to improve the performance of tree-based methods. This could involve experimenting with multivariate tree-based methods, like the multivariate random forest (Sega & Xiao, 2011). Tree-based methods with random effects have been developed (Fokkema, Edbrooke-Childs, & Wolpert, 2020; Fokkema, Smits, Zeileis, Hothorn, & Kelderman, 2018; Hajjem, Bellavance, & Larocque, 2014; Hajjem, Larocque, & Bellavance, 2017; Sela & Simonoff, 2012), but still have to be adapted to missing data. Another, simpler approach could be to average all data at Level 2, followed by imputation and then integration back into the full dataset. Such an approach could improve the effectiveness of tree-based methods in dealing with hierarchical data.

In conclusion, our study underscores the continued effectiveness of MICE in dealing with hierarchical data, particularly in terms of rejection rates. However, the emerging tree-based methods, especially `mixgb`, show potential for bias reduction, suggesting their usefulness as alternatives in certain contexts. This dual finding opens new avenues for future research and practical applications in data imputation, highlighting the dynamic nature of the field.

Acknowledgement

The authors gratefully acknowledge the computing time provided on the Linux HPC cluster at Technical University Dortmund (LiDO3), partially funded in the course of the Large-Scale Equipment Initiative by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) as project 271512359.

Funding

The project "From Prediction to Agile Interventions in the Social Sciences (FAIR)" is receiving funding from the programme "Profilbildung 2020", an initiative of the Ministry of Culture and Science of the State of Northrhine Westphalia. The sole responsibility for the content of this publication lies with the authors.

Contributions

Author contributions: **Conceptualization:** JS, PD; **Data curation:** JS, KG; **Formal analysis:** JS, KG; **Funding acquisition:** PD; **Investigation:** JS, KG; **Methodology:** JS, KG, PD; **Project administration:** JS; **Resources:** ; **Software:** KG; **Supervision:** PD; **Validation:** JS, KG; **Visualization:** JS, KG; **Writing – original draft:** JS, KG; **Writing – review & editing:** JS, KG, PD

References

Aitkin, M., & Longford, N. (1986). Statistical modelling issues in school effectiveness studies. *Journal of the Royal Statistical Society: Series A (General)*, *149*(1), 1-26. doi:https://doi.org/10.2307/2981882

Audigier, V., White, I. R., Jolani, S., Debray, T. P. A., Quartagno, M., Carpenter, J., ... Resche-Rigon, M. (2018). Multiple Imputation for Multilevel Data with Continuous and Binary Variables. *Statistical Science*, *33*(2), 160-183. doi:10.1214/18-STS646

Breiman, L. (2001). Random forest missing data algorithms. *Machine Learning*, *45*, 5-32. doi:10.1023/A:1010933404324

Breiman, L. (2003). *Manual: Setting up, using, and understanding random forests V4.0* (Tech. Rep.). Retrieved from `https://www.stat.berkeley.edu/~breiman/Using_random_forests_v4.0.pdf`

Cameron, R. C., & Miller, D. L. (2015). A Practitioner's Guide to Cluster-Robust Inference. *Journal of Human Resources*, *50*, 317–372. doi:10.3368/jhr.50.2.317

Chen, T., & Guestrin, C. (2016). XGBoost: A scalable tree boosting system. *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, *13-17-August*, 785–794. doi:10.1145/2939672.2939785

Deng, Y. (2023). mixgb: Multiple imputation through 'xgboost' [Computer software manual]. (R package version 1.0.2) Retrieved from https://CRAN.R-project.org/package=mixgb

Deng, Y., & Lumley, T. (online first). Multiple imputation through XGBoost. *Journal of Computational and Graphical Statistics*, 19 p. doi:10.1080/10618600.2023.2252501

Enders, C. K., Mistler, S. A., & Keller, B. T. (2016). Multilevel multiple imputation: A review and evaluation of joint modeling and chained equations imputation. *Psychological Methods*, *21*(2), 222-240. doi:10.1037/met0000063

Fahrmeir, L., Kneib, T., Lang, S., & Marx, B. (2013). *Regression: Models, methods and applications*. Berlin: Springer.

Fokkema, M., Edbrooke-Childs, J., & Wolpert, M. (2020). Generalized linear mixed-model (GLMM) trees: A flexible decision-tree method for multilevel and longitudinal data. *Psychotherapy Research*, 1–13. doi:10.1080/10503307.2020.1785037

Fokkema, M., Smits, N., Zeileis, A., Hothorn, T., & Kelderman, H. (2018). Detecting treatment-subgroup interactions in clustered data with generalized linear mixed-effects model trees. *Behavior Research Methods*, *50*(5), 2016–2034. doi:10.3758/s13428-017-0971-x

Gałecki, A., & Burzykowski, T. (2013). *Linear mixed-effects models using R: A step-by-step approach*. New York: Springer.

Geurts, P., Ernst, D., & Wehenkel, L. (2006). Extremely randomized trees. *Machine Learning*, *63*, 3-42. doi:10.1007/s10994-006-6226-1

Graham, J. W. (2012). *Missing data. analysis and design*. New York: Springer.

Grund, S., Lüdtke, O., & Robitzsch, A. (2024). Missing data in the analysis of multilevel and dependent data. In M. Stemmler, W. Wiedermann, & F. Huang (Eds.), *Dependent data in social sciences research: Forms, issues, and methods of analysis* (2nd ed.). Springer.

Grund, S., Lüdtke, O., & Robitzsch, A. (2018a). Multiple imputation of missing

data at level 2: A comparison of fully conditional and joint modeling in multilevel designs. *Journal of Educational and Behavioral Statistics*, *43*(3), 316-353. doi:10.3102/1076998617738087

Grund, S., Lüdtke, O., & Robitzsch, A. (2018b). Multiple imputation of missing data for multilevel models: Simulations and recommendations. *Organizational Research Methods*, *21*(1), 111-149. doi:10.1177/1094428117703686

Hajjem, A., Bellavance, F., & Larocque, D. (2014). Mixed-effects random forest for clustered data. *Journal of Statistical Computation and Simulation*, *84*(6), 1313–1328. doi:10.1080/00949655.2012.741599

Hajjem, A., Larocque, D., & Bellavance, F. (2017). Generalized mixed effects regression trees. *Statistics and Probability Letters*, *126*, 114–118. doi:10.1016/j.spl.2017.02.033

Heymann, M. (2017). 1970s: Turn of an era in the history of science? *Centaurus*, *59*(1-2), 1-9. doi:https://doi.org/10.1111/1600-0498.12146

Hox, J. (1998). Multilevel modeling: When and why. In I. Balderjahn, R. Mathar, & M. Schader (Eds.), *Classification, data analysis, and data highways* (pp. 147–154). Berlin: Springer.

Hox, J., & Roberts, J. K. (2010). *Handbook of advanced multilevel analysis* (First ed.). New York: Routledge. doi:10.4324/9780203848852

Levi, S., Wolf, I., Sommer, S., & Howe, P. D. (2023). Local support of climate change policies in Germany over time. *Environmental Research Letters*, *18*(6), 1–19. doi:10.1088/1748-9326/acd406

Little, R. J., & Rubin, D. B. (2002). *Statistical analysis with missing data* (Second ed.). Hoboken, NJ: Wiley & Sons.

Lüdtke, O., Robitzsch, A., & Grund, S. (2017). Multiple imputation of missing data in multilevel designs: A comparison of different strategies. *Psychological Methods*, *22*(1), 141–165. doi:10.1037/met0000096

Mayer, M. (2021). missranger: Fast imputation of missing values [Computer software manual]. (R package version 2.1.3) Retrieved from `https://CRAN.R-project.org/package=missRanger`

Pinheiro, J. C., & Bates, D. M. (2004). *Mixed-effects models in S and S-Plus*. New

York: Springer.

Ramosaj, B., Amro, L., & Pauly, M. (2020). A cautionary tale on using imputation methods for inference in matched-pairs design. *Bioinformatics*, *36*(10), 3099-3106. doi:10.1093/bioinformatics/btaa082

Rubin, D. B. (1976). Inference and missing data. *Biometrika*, *63*(3), 581-592. doi:10.1093/biomet/63.3.581

Rubin, D. B. (1987). *Multiple imputation for nonresponse in surveys* (Second ed.). USA: Wiley & Sons.

Sajeev, S., Champion, S., Maeder, A., & Gordon, S. (2022). Machine learning models for identifying pre-frailty in community dwelling older adults. *BMC Geriatrics*, *22*(1), 1–12. doi:10.1186/s12877-022-03475-9

Schafer, J. L. (2000). *Analysis of incomplete multivariate data*. Boca Raton: Chapman & Hall/CRC.

Schafer, J. L., & Olsen, M. K. (1998). Multiple imputation for multivariate missing-data problems: A data analyst's perspective. *Multivariate Behavioral Research*, *33*(4), 545-571. (PMID: 26753828) doi:10.1207/s15327906mbr3304_5

Schwerter, J., Bleher, J., Doebler, P., & McElvany, N. (2023). Metropolitan, urban, and rural regions–how regional differences affect elementary school students' academic achievement, well-being, and motivation. *Working paper*. doi:https://dx.doi.org/10.2139/ssrn.4368170

Schwerter, J., Gurtskaia, K., Romero, A., Zeyer-Gliozzo, B., & Pauly, M. (2023). Evaluating tree-based imputation methods as an alternative to MICE PMM for drawing inference in empirical studies. *Working paper*. doi:https://doi.org/10.48550/arXiv.2401.09602

Sega, M., & Xiao, Y. (2011). Multivariate random forests. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, *1*(1), 80–87. doi:10.1002/widm.12

Sela, R. J., & Simonoff, J. S. (2012). RE-EM trees: A data mining approach for longitudinal and clustered data. *Machine Learning*, *86*(2), 169–207. doi:10.1007/s10994-011-5258-3

Steenbergen, M. R., & Jones, B. S. (2002). Modeling multilevel data

structures. *American Journal of Political Science*, *46*(1), 218-237. doi:https://doi.org/10.2307/3088424

Stekhoven, D. J., & Bühlmann, P. (2012). Missforest—non-parametric missing value imputation for mixed-type data. *Bioinformatics*, *28*(1), 112-118. doi:10.1093/bioinformatics/btr597

Suh, H., & Song, J. (2023). A comparison of imputation methods using machine learning models. *Communications for Statistical Applications and Methods*, *30*(3), 331–341. doi:10.29220/CSAM.2023.30.3.331

Tang, F., & Ishwaran, H. (2017). Random forest missing data algorithms. *Statistical Analysis and Data Mining: The ASA Data Science Journal*, *10*(6), 363-377. doi:10.1002/sam.11348

Thurow, M., Dumpert, F., Ramosaj, B., & Pauly, M. (2021). Goodness (of fit) of Imputation Accuracy: The GoodImpact Analysis. *Working paper*. doi:https://doi.org/10.48550/arXiv.2101.07532

van Buuren, S., & Groothuis-Oudshoorn, K. (2011). mice: Multivariate imputation by chained equations in R. *Journal of Statistical Software*, *45*(3), 1-67. doi:10.18637/jss.v045.i03

van der Heijden, G. J., Donders, A. R. T., Stijnen, T., & Moons, K. G. (2006). Imputation of missing values is superior to complete case analysis and the missing-indicator method in multivariable diagnostic research: A clinical example. *Journal of Clinical Epidemiology*, *59*(10), 1102-1109. doi:https://doi.org/10.1016/j.jclinepi.2006.01.015

van Buuren, S. (2011). Multiple imputation of multilevel data. In J. Hox & J. Roberts (Eds.), *Handbook of advanced multilevel analysis* (pp. 173–196). New York: Routledge.

van Buuren, S. (2012). *Flexible imputation of missing data*. Boca Raton: CRC Press.

van Buuren, S., Brand, J., Groothuis-Oudshoorn, C., & Rubin, D. (2006). Fully conditional specification in multivariate imputation. *Journal of Statistical Computation and Simulation*, *76*(12), 1049-1064. doi:10.1080/10629360600810434

van Buuren, S., & Groothuis-Oudshoorn, K. (2011). mice: Multivariate imputation by chained equations in R. *Journal of Statistical Software*, *45*(3), 1–67.

doi:10.18637/jss.v045.i03

van Buuren, S., & Oudshoorn, K. (1999). *Flexible multivariate imputation by mice.* Leiden: TNO.

Waller, N. G. (2022). fungible: Psychometric functions from the waller lab. [Computer software manual]. (version 2.2.1)

Wright, M. N., & Ziegler, A. (2017). ranger: A Fast Implementation of Random Forests for High Dimensional Data in C++ and R. *Journal of Statistical Software*, *77*(1). doi:10.18637/jss.v077.i01

Wright, M. N., Ziegler, A., & König, I. R. (2016). Do little interactions get lost in dark random forests? *BMC Bioinformatics*, *17*, 1–10.

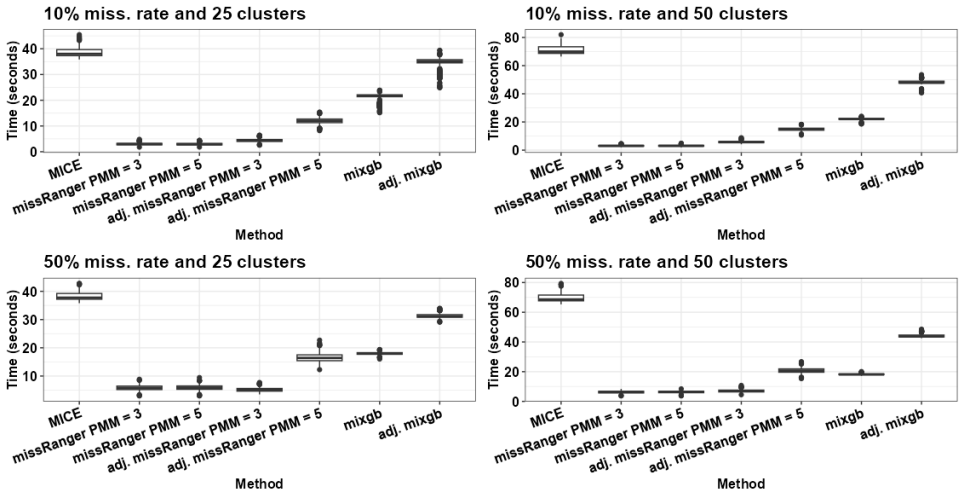
Appendix A. Running times

To also compare the computational time costs for the methods used, Figure A1 illustrates the box plots of the imputation running times for each method. Among all methods MICE is the slowest under all designs. Dummy-adjusted XGBoost is only slightly faster than MICE for data with 25 clusters and almost twice as fast as MICE with 50 clusters. The standard implementations of `missRanger` are always the fastest (almost always ten times faster than MICE). All methods take more time imputing data with 50 clusters than 25 clusters.

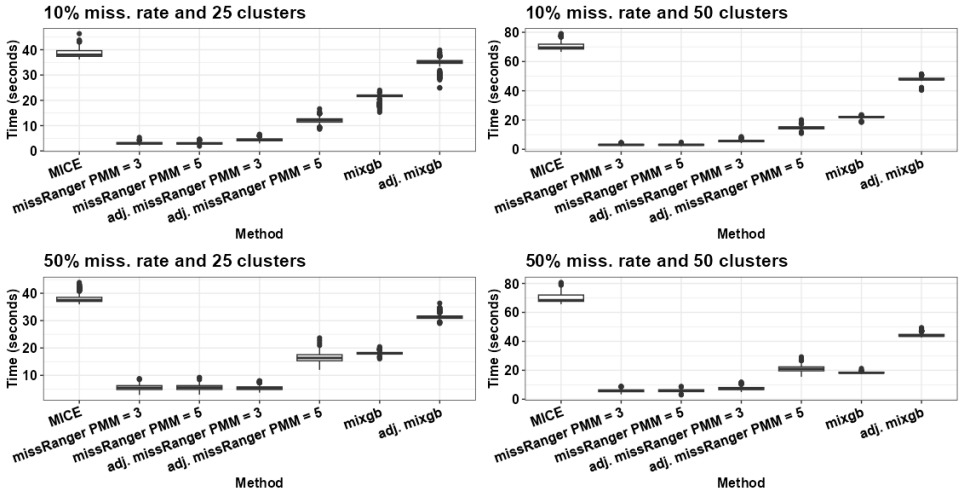
As for random slopes designs, MICE once again has the longest running times as shown in Figure A2. Dummy-adjusted XGBoost takes almost as much time as MICE with 10% missingness and 25 clusters. All the other methods are at least twice as fast. Especially standard `missRanger`, which is around eight times faster than MICE. The standard XGBoost takes more than twice as much time as standard `missRanger`.

Figure A1.: Running Times for random intercept model

(a) MCAR



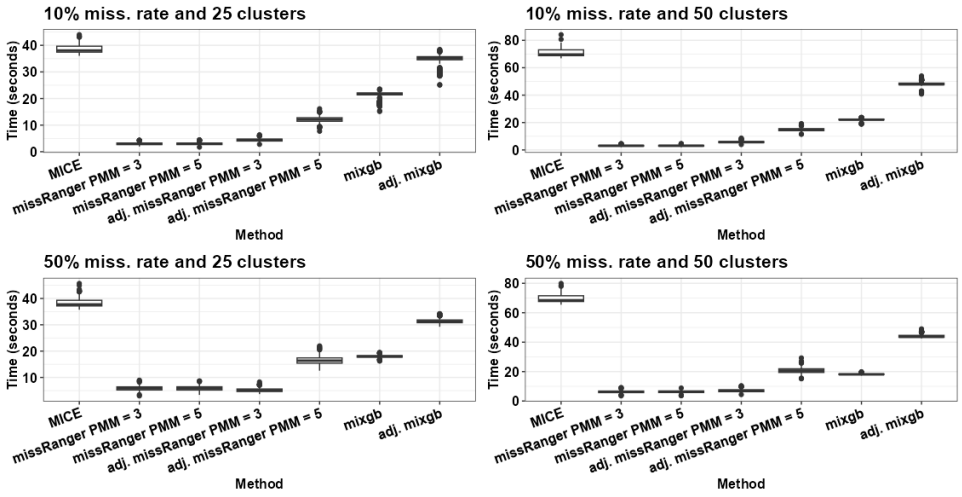
(b) MAR



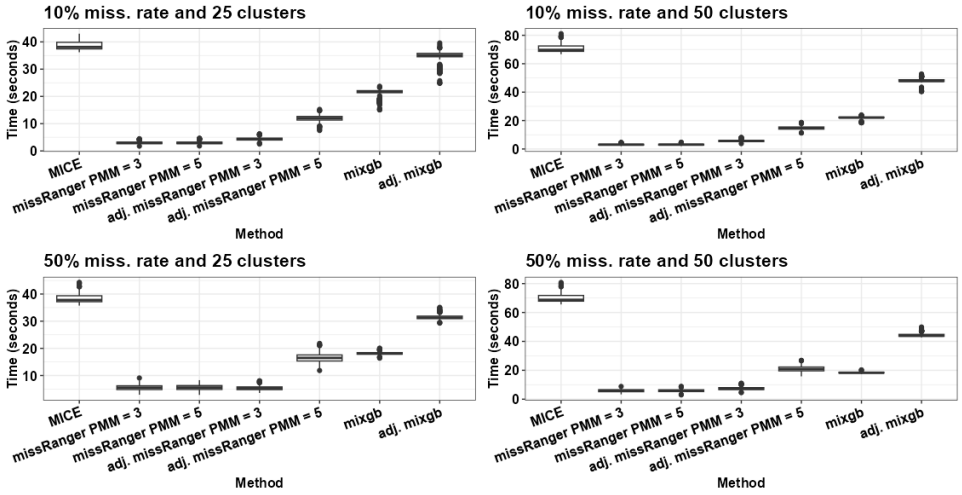
Running Times for random intercept designs with 10% (first rows) and 50% (second rows) missingness and 25 (first columns) and 50 (second columns) clusters

Figure A2.: Running Times for random slope models

(a) MCAR



(b) MAR



Imputation Running Times for random intercept designs with 10% (first rows) and 50% (second rows) missingness and 25 (first columns) and 50 (second columns) clusters