The Missing Indicator Method: From Low to High Dimensions

Mike Van Ness
Stanford University
Stanford, CA, USA

Tomas M. Bosschieter
Stanford University
Stanford, CA, USA

Roberto Halpin-Gregorio
Cornell University
Ithaca, NY, USA

Madeleine Udell
Stanford University
Stanford, CA, USA

ABSTRACT

Missing data is common in applied data science, particularly for tabular data sets found in healthcare, social sciences, and natural sciences. Most supervised learning methods only work on complete data, thus requiring preprocessing such as missing value imputation to work on incomplete data sets. However, imputation alone does not encode useful information about the missing values themselves. For data sets with informative missing patterns, the Missing Indicator Method (MIM), which adds indicator variables to indicate the missing pattern, can be used in conjunction with imputation to improve model performance. While commonly used in data science, MIM is surprisingly understudied from an empirical and especially theoretical perspective. In this paper, we show empirically and theoretically that MIM improves performance for informative missing values, and we prove that MIM does not hurt linear models asymptotically for uninformative missing values. Additionally, we find that for high-dimensional data sets with many uninformative indicators, MIM can induce model overfitting and thus test performance. To address this issue, we introduce Selective MIM (SMIM), a novel MIM extension that adds missing indicators only for features that have informative missing patterns. We show empirically that SMIM performs at least as well as MIM in general, and improves MIM for high-dimensional data. Lastly, to demonstrate the utility of MIM on real-world data science tasks, we demonstrate the effectiveness of MIM and SMIM on clinical tasks generated from the MIMIC-III database of electronic health records.

KEYWORDS

missing values, supervised learning, imputation, healthcare

1 INTRODUCTION

Missing data is an unavoidable consequence of tabular data collection in many domains. Nonetheless, most statistical studies and machine learning algorithms not only assume complete data, but cannot run on data sets with missing entries. Two common preprocessing methods are used to address this issue. The first method is complete case analysis, in which the fully observed data samples are used, while the others are discarded. However, discarding data reduces statistical power and can bias results, as partially-observed data samples may still contain important information. Furthermore, sometimes *all* data samples have missing values, in which case discarding incomplete samples is catastrophic.

The second method is missing value imputation, in which missing values are replaced by estimates from the observed data. While missing value imputation is well studied and widely used [28, 48], it comes with two problems. Firstly, most imputation methods are only (provably) effective when missing values are missing at random (MAR), which is often violated in real-world data, e.g. medical data [34]. In particular, many real-world data sets exhibit *informative missingness*, i.e. when the pattern of missing data encodes information about the response variable. When data has informative missingness, the MAR assumption is not typically satisfied, making imputation methods less effective.

Secondly, imputation methods are typically evaluated by their reconstruction error. This method of evaluation is reasonable when the task of interest itself is imputation, e.g. for recommender systems. However, optimal imputation in terms of lowest reconstruction error is not necessarily required for optimizing downstream prediction accuracy. In fact, recent theoretical work has shown that simple imputation schemes can still result in Bayes optimal predictions if the imputation function is paired with an appropriate prediction function [4, 21, 25]. Thus, it is possible to obtain optimal predictions without accurate imputations, which is particularly useful when making accurate imputations is challenging, i.e. for MNAR data.

An alternative strategy, particularly when missing values are informative, is the Missing Indicator Method (MIM), which directly leverages the signal in the missingness itself rather than optimizing imputation accuracy. For each partially observed feature, MIM adds a corresponding indicator feature to indicate whether the feature is missing or not. MIM can be used with any imputation method, but is most commonly used with mean imputation, which is equivalent to 0-imputation after centering features. MIM is a common method in practice, e.g. it is implemented in scikit-learn [33], yet MIM remains understudied from an empirical and, in particular, theoretical perspective.

In this paper, we show both theoretically and empirically that MIM is an effective strategy for supervised learning in a wide variety of data regimes. While some previous work has studied MIM related to statistical inference [15, 22], our work focuses on MIM as part of a supervised learning pipeline, where prediction accuracy is the primary objective opposed to inference. Additionally, to better handle high-dimensional data sets where extra uninformative indicator features can lead to overfitting, we introduce *Selective MIM* (SMIM), a novel MIM extension to adaptively select which indicators are useful to add to the data set.

Our main contributions are as follows:

- We provide a novel theoretical analysis of MIM for linear regression. In particular, we prove that MIM successfully encodes the signal in missing values when the missingness is informative, thus increasing prediction performance. Further, when missingness is not informative, MIM does not degrade performance asymptotically.

- We introduce *Selective MIM* (SMIM), a novel improvement to MIM that uses significance tests to select only the informative indicators. SMIM stabilizes MIM for high-dimensional data sets, where MIM can cause overfitting and thus increased variance in training.
- We conduct an extensive empirical study of MIM and SMIM on synthetic data and real-world data, for various different imputation and prediction models. We show that MIM plus mean imputation is a strong method while being much more efficient than more complicated imputation schemes. Additionally, we show that SMIM performs at least as well as MIM in general, while outperforming MIM on high-dimensional data sets.
- To demonstrate the utility of MIM and SMIM in real-world applications, we evaluate these methods on the MIMIC-III data set [20], a clinical data set of electronic health records (EHRs). We show that (S)MIM improves model performance on a collection of clinical tasks, showing that real-world data often has informative missing values with signal that can be captured with MIM.

2 MISSING VALUES IN SUPERVISED LEARNING

In this section we introduce the problem statement and relevant work, and expand on how we build on existing work.

2.1 Notation

We use the following conventions. Upper case letters X are random variables. Bold letters $\boldsymbol{X}$ are vectors, which by default are always column vectors. Subscripts X_j denote components of $\boldsymbol{X}$, and superscripts $\boldsymbol{X}^{(i)}$ denote sample vectors in a data set. We use n for the number of samples/rows, and p for the number of features/columns. We reserve i as an index on samples and j as an index on features. $\boldsymbol{R}$ will denote indicator features corresponding to missing values in $\boldsymbol{X}$. The sets $\text{obs}(\boldsymbol{R}) = \{j : R_j = 0\}$ and $\text{miss}(\boldsymbol{R}) = \{j : R_j = 1\}$ indicate observed and missing components of a vector. $X_j \perp\!\!\!\perp X_k$ means X_j and X_k are independent as random variables.

2.2 Problem Statement

We consider the following supervised learning setup. Let $\boldsymbol{X} \in \mathbb{R}^p$ be a vector of p features, and $Y \in \mathcal{Y}$ a response to be predicted from $\boldsymbol{X}$, where $\mathcal{Y} = \mathbb{R}$ for regression tasks and $\mathcal{Y} = \{0, 1, \ldots, k-1\}$ for k-class classification tasks. The vector $\boldsymbol{X}$ contains a complete set of values, in the sense that a value exists for each component of $\boldsymbol{X}$ even if that value is not observed. In reality, we observe $\boldsymbol{Z} \in \{\mathbb{R} \cup \{*\}\}^p$ where $Z_j = X_j$ when X_j is observed and $Z_j = `*'$ when X_j is unobserved/missing. $\boldsymbol{Z}$ yields the random binary vector $\boldsymbol{R} \in \{0, 1\}^p$ that indicates the missing and observed components of $\boldsymbol{Z}$. Specifically, $R_j = 0$ when X_j is observed and $R_j = 1$ when X_j is missing.

Given a loss function $\mathcal{L} : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}$ and a distribution over pairs $(\boldsymbol{Z}, Y) \sim \mathcal{D}$, the goal is to find function $f : \{\mathbb{R} \cup \{*\}\}^p \to \mathcal{Y}$ that minimizes the expected loss:

$$f^* = \underset{f}{\text{argmin}}\ \mathbb{E}_{\mathcal{D}}\left[\mathcal{L}(f(\boldsymbol{Z}), Y)\right]. \tag{1}$$

We will often drop the subscript $\mathcal{D}$ from expectations with the default being that the expectation is with respect to $\mathcal{D}$. The function f can be a pipeline that combines an imputation method with a prediction method, or can directly predict Y from $\boldsymbol{Z}$ without imputation, e.g. fitting a different model per missing pattern.

2.3 Types of Missing Data

Traditionally, partially observed data is categorized by the missing mechanism, i.e. the distribution of $\boldsymbol{R}$ conditional on the complete data $\boldsymbol{X}$. The most typical categorization of missing mechanisms is [30]:

- **Missing Completely at Random (MCAR).** The missing pattern is random and independent of the data: $P(\boldsymbol{R} \mid \boldsymbol{X}) = P(\boldsymbol{R})$.
- **Missing at Random (MAR).** The missing pattern is independent of unobserved values $\boldsymbol{X}_{\text{miss}(\boldsymbol{R})}$ conditioned on observed values $\boldsymbol{X}_{\text{obs}(\boldsymbol{R})}$: $P(\boldsymbol{R} \mid \boldsymbol{X}) = P(\boldsymbol{R} \mid \boldsymbol{X}_{\text{obs}(\boldsymbol{R})})$.
- **Missing Not at Random (MNAR).** The missing pattern may depend on unobserved values.

The MCAR and MAR mechanisms are sometimes considered *ignorable* in the sense that these mechanisms may be ignored for likelihood-based inference [38]. Many imputation methods work well on MCAR or MAR data [18, 45, 49, 56], while MNAR data is more challenging for imputation methods and requires modeling the missing mechanism itself [23, 31, 44].

While missing mechanisms define the relationship between $\boldsymbol{R}$ and $\boldsymbol{X}$, informative missingness is defined by the relationship between $\boldsymbol{R}$ and Y:

Definition 2.1. *For $(\boldsymbol{Z}, Y) \sim \mathcal{D}$ with missing pattern $\boldsymbol{R}$, the missing pattern is* informative *if $\boldsymbol{R} \not\perp\!\!\!\perp Y$.*

Assuming $\boldsymbol{X} \not\perp\!\!\!\perp Y$, this definition implies that MAR and MNAR data always have informative missingness. MCAR data can also have informative missingness if $\boldsymbol{R} \not\perp\!\!\!\perp Y$ despite $\boldsymbol{R} \perp\!\!\!\perp \boldsymbol{X}$, although this is rare in practice. In previous literature, informative missingness is most often associated with MNAR data, as in this setting the missingness is most informative [27, 39]. In general, we expect MIM to improve predictions when missingness is informative. Yet, we can have MAR data and not see any benefit to MIM over imputation alone if $\boldsymbol{R} \perp\!\!\!\perp Y \mid \boldsymbol{X}_{obs}$

2.4 Related Work

Missing values have been an important topic of study in statistics for decades. Classical work focused on statistical inference problems [8, 30, 38], while more recent work has focused more on imputation, ranging from statistical methods [18, 55] to iterative approaches [45, 49] and deep learning [14, 31, 54].

Even though MIM is commonly used in practice (e.g. it is implemented in sklearn [33]), its properties are surprisingly understudied in statistical and machine learning literature, particularly from the perspective of supervised learning. MIM has been explored more in the medical literature due to the ubiquity of missing values in medical data sets, and particularly the tendency of these missing values to be informative. A few medical papers advocate for MIM with mean imputation [41, 43], and some for MIM with other imputation methods [36, 42]. Other medical papers caution the use of MIM when statistical estimation and inference is the downstream task, as then using MIM can add bias to parameter estimates [15, 22]. We instead focus on the problem of optimal supervised learning.

Many previous methodological and empirical works have studied missing value preprocessing for supervised learning, yet most do not include MIM or a similar method to capture informative missingness. Some empirical studies have evaluated combinations of imputation and prediction models [11, 52], but these most often exclude MIM. Recent AutoML methods attempt to optimize the entire supervised learning pipeline [10, 12, 53], but without considering MIM. Deep learning can be used to jointly optimize imputation and prediction if both models are neural networks [19, 24], but these methods do not capture informative missingness without additionally using missing indicators. Decision trees can uniquely handle missing values without imputation, e.g. using the Missing Incorporated as Attribute method [21, 47], but such methods are only applicable to tree-based methods.

Lastly, some recent papers have explored impute-then-predict pipelines from a theoretical perspective. [4, 21, 25] show that accurate imputation is not needed to produce Bayes optimal predictions as long as a powerful enough predictor is used. While this theory is not immediately applicable in practice, since constructing these predictors would often be impossible, it does give motivation for developing missing value preprocessing methods that do not focus on imputation accuracy, such as MIM with mean imputation. Perhaps the most relevant previous work to this paper is [26], which gives a risk bounds for linear models with 0-imputation using both MIM and an expanded linear model. We also present theory for MIM with linear models in this paper, but under different assumptions and focusing on asymptotics rather than risk bounds. We also perform a much more diverse set of empirical experiments in this paper compared to [26].

3 MISSING INDICATOR METHOD THEORY

In this section, we present a theoretical study of MIM for linear regression. Following the notation from Section 2, let $\boldsymbol{X} \in \mathbb{R}^p$ be p features, $Y \in \mathbb{R}$ a continuous response, and $\boldsymbol{Z} \in \{\mathbb{R} \cup \{*\}\}^p$ the observed features with potentially missing values. The full optimization problem in Eq. (1) with squared error loss $\mathcal{L}(x, y) = (x - y)^2$ is solved by the conditional expectation $\mathbb{E}[Y \mid \boldsymbol{Z}]$. However, as explained in [24, 26], this conditional expectation is combinatorial in nature, as it involves estimating a different expectation for each missing pattern $\boldsymbol{R}$:

$$f^* = \mathbb{E}[Y \mid \boldsymbol{Z}] = \mathbb{E}[Y \mid \boldsymbol{X}_{\text{obs}(\boldsymbol{R})}, \boldsymbol{R}] \tag{2}$$

$$= \sum_{\boldsymbol{r} \in \{0,1\}^d} \mathbb{E}[Y \mid \boldsymbol{X}_{\text{obs}(\boldsymbol{r})}, \boldsymbol{R} = \boldsymbol{r}] \mathbb{1}_{\boldsymbol{R}=\boldsymbol{r}}. \tag{3}$$

In [24, 26], the conditional expectation in Eq. (2) is estimated after assuming that $\boldsymbol{X}$ comes from a Gaussian distribution. In this paper, we make no distributional assumptions, and instead make the following assumptions:

Assumption 3.1. *Y is centered as $Y \leftarrow Y - \mathbb{E}[Y]$, while $\boldsymbol{Z}$ is centered over the observed entries $Z_j \leftarrow Z_j - \mathbb{E}[Z_j \mid R_j = 0]$ and imputed with 0, resulting in the imputed data vector $\tilde{\boldsymbol{Z}}$ defined as*

$$\tilde{Z}_j = \begin{cases} X_j & R_j = 0, \\ 0 & R_j = 1. \end{cases} \tag{4}$$

Assumption 3.2. *f is constrained to be a linear function, and thus f^* is the best linear prediction function.*

We now examine f^* with and without MIM. Specifically, we compare the best linear predictors (BLPs):

$$\boldsymbol{\beta}^* = \underset{\boldsymbol{\beta}}{\operatorname{argmin}}\, \mathbb{E}[(Y - \tilde{\boldsymbol{Z}}^T \boldsymbol{\beta})^2], \tag{5}$$

$$\boldsymbol{\beta}^*_{MIM}, \boldsymbol{\gamma}^*_{MIM} = \underset{(\boldsymbol{\beta},\boldsymbol{\gamma})}{\operatorname{argmin}}\, \mathbb{E}[(Y - \tilde{\boldsymbol{Z}}^T \boldsymbol{\beta} - \boldsymbol{R}^T \boldsymbol{\gamma})^2]. \tag{6}$$

In practice, we would estimate these BLPs using a training set $\{(\boldsymbol{Z}^{(1)}, \boldsymbol{R}^{(1)}), \ldots (\boldsymbol{Z}^{(n)}, \boldsymbol{R}^{(n)})\}$, obtaining the standard ordinary least squares (OLS) estimates $\hat{\boldsymbol{\beta}}$, $\hat{\boldsymbol{\beta}}_{MIM}$, and $\hat{\boldsymbol{\gamma}}_{MIM}$. Under mild conditions, most notably that $\mathbb{E}[\tilde{\boldsymbol{Z}}\tilde{\boldsymbol{Z}}^T]$ and $\mathbb{E}[(\tilde{\boldsymbol{Z}}^T, \boldsymbol{R}^T)^T (\tilde{\boldsymbol{Z}}^T, \boldsymbol{R}^T)]$ are invertible, these OLS estimates will converge to the corresponding BLPs in Eqs. (5) and (6); see [16] for further details. This convergence validates the study of the best linear predictors, as they serve as the limits of the OLS estimates obtained in practice.

Theorem 3.1 starts by analyzing the simple case when $p = 1$, which has particularly nice properties.

Theorem 3.1. *Grant Assumptions 3.1 and 3.2. $p = 1$, and let $\mathcal{M} = \{i \colon R^{(i)} = 1\}$ be the missing samples, then the OLS estimates are*

$$\hat{\beta}_{MIM} = \hat{\beta}, \quad \hat{\gamma}_{MIM} = \frac{1}{|\mathcal{M}|} \sum_{i \in \mathcal{M}} Y^{(i)}, \tag{7}$$

and thus the BLPs are

$$\beta^*_{MIM} = \beta^*, \quad \gamma^*_{MIM} = \mathbb{E}[Y \mid R = 1]. \tag{8}$$

A direct corollary of Theorem 3.1 is that if R is uninformative, then

$$\gamma^*_{MIM} = \mathbb{E}[Y \mid R = 1] = \mathbb{E}[Y] = 0 \tag{9}$$

since Y is centered. On the other hand, if R is informative, then $\gamma^*_{MIM} = \mathbb{E}[Y \mid R = 1] \neq 0$, allowing the model to adjust its predictions when X is missing. Specifically, because X is imputed with 0, the best linear prediction function is

$$f^*_{linear}(X) = \begin{cases} X\beta^* & \text{if } R = 0, \\ \mathbb{E}[Y \mid R = 1] & \text{if } R = 1. \end{cases} \tag{10}$$

or correspondingly in finite sample

$$\hat{f}_{linear}(X) = \begin{cases} X\hat{\beta} & \text{if } R = 0, \\ \frac{1}{|\mathcal{M}|} \sum_{i \in \mathcal{M}} Y^{(i)} & \text{if } R = 1. \end{cases} \tag{11}$$

We see that when X is observed, the model ignores R as a feature, and when X is missing, the model predicts the average of Y among the missing values. Note that when $p = 1$, this result occurs both in finite sample and asymptotically.

We now consider the more general case of $p > 1$ features in Theorem 3.2.

Theorem 3.2. *Grant Assumptions 3.1 and 3.2.*

(a) If the missing mechanism is MCAR and $\boldsymbol{R}$ is uninformative, then

$$\boldsymbol{\beta}^* = \boldsymbol{\beta}^*_{MIM}, \quad \boldsymbol{\gamma}^* = 0. \tag{12}$$

(b) If (i) the missing mechanism is self-masking, i.e. $P(\boldsymbol{R} \mid \boldsymbol{X}) = \prod_j P(R_j \mid X_j)$*; (ii)* $X_j \perp\!\!\!\perp X_k$ *for* $j \neq k$*; and (iii)* $\boldsymbol{R}$ *is centered, then*

$$\boldsymbol{\beta}^* = \boldsymbol{\beta}^*_{MIM}, \tag{13}$$

$$\tag{14}$$

and for $j = 1, \ldots, p$

$$\gamma^*_{MIM_j} = \mathbb{E}[Y \mid X_j \textit{ missing}] - \mathbb{E}[Y \mid X_j \textit{ observed}]. \tag{15}$$

(c) If $\{1, \ldots, p\}$ *can be partitioned into* d *blocks* $B_1, \ldots, B_d$ *with block-independence:* $R_j, X_j \perp\!\!\!\perp R_k, X_k$ *when* j, k *are in different blocks, then for* $j \in B_m$

$$\beta^*_{MIM_j} = \sum_{\ell \in B_m} a\mathbb{E}[\tilde{Z}_j Y] + b(\mathbb{E}[Y \mid X_\ell \textit{ missing}] - \mathbb{E}[Y \mid X_\ell \textit{ observed}]) \tag{16}$$

for a *and* b *are functions of* $(\tilde{\boldsymbol{Z}}, \boldsymbol{R})$*, and the same for* $\gamma^*_{MIM_j}$ *but for different* a *and* b*.*

The proofs of Theorems 3.1 and 3.2 are given in Appendix A. Theorem 3.2 (a) shows that when the missing mechanism is MCAR and $\boldsymbol{R}$ is uninformative, the best linear predictions are the same regardless of whether MIM is used or not. Since MIM is not expected to improve model performance when missingness is not informative, it is promising that MIM does not decrease the accuracy of linear models in this setting, at least asymptotically. This is important because practitioners are often unaware of the missing mechanism of the data, which is difficult to test [1]. It is therefore crucial to understand the effects of MIM across all missing mechanisms. This evidence that MIM does not hurt performance even in a worst-case scenario justifies its use when predictive performance is prioritized.

Parts (b) and (c) of Theorem 3.2 show that when missingness is informative, the best linear prediction function using MIM learns how the missing values impact the distribution of Y, thereby leveraging the signal in the missing values. In part (b), under the fully-independent self-masking mechanism, the form of $\boldsymbol{\gamma}^*_{MIM}$ in Eq. (15) directly accounts for the informative missingness in each feature, i.e. $\mathbb{E}[Y \mid X_j \text{ missing}] - \mathbb{E}[Y \mid X_j \text{ observed}]$. Part (c) presents a more general case, where independence is only assumed blockwise. Nonetheless, the formula for the BLP in (c) still contains the same expression as in Eq. (15) from (b), which encodes the (potential) informative missingness of each feature.

As discussed in Section 2.4, even though MIM is commonly used in practice, very little previous work has studied MIM theoretically, especially from the lens of supervised learning. A notable exception is [26], in particular Theorem 5.2, which presents an upper bound for the risk of the linear model with MIM, i.e. the expected loss in Eq. (6). [26] compares this risk bound to the risk bound for the "expanded" linear model, which fits a separate linear model for each missing pattern in Eq. (3). Theorem 3.2 in our paper is complementary to these results, as our results specify the values for the best linear predictors under specific missing mechanisms. While the risk bounds in [26] are useful for comparing the finite-sample risk of MIM compared to other models, our results present a more practical understanding of linear model coefficients with MIM, and how these coefficients can directly encode or ignore missingness depending on the missing mechanism.

4 SELECTIVE MIM

Observe that MIM adds a missing indicator for *every* feature that is partially observed, regardless of the missing mechanism. Theorem 3.2 (a) justifies this behavior asymptotically, since the uninformative indicators can be ignored as $n \to \infty$. However, in finite sample, and particularly for high-dimensional data, adding many uninformative indicators can cause overfitting and thus can hurt model performance.

Keeping only the informative indicators from MIM would help stabilize MIM on high-dimensional data. One strategy to do such selection would be to use a statistical test to discover the missing mechanism, and use this information to determine which indicators to keep. However, testing for the missing mechanism is challenging. Previous literature has suggested tests for MCAR, notably Little's MCAR test [29], but these tests have been criticized [1]. Testing for MAR or MNAR, on the other hand, is impossible, since it would require access to the unobserved data. From Definition 2.1, however, we can instead test for informative missingness directly by testing the relationship between $\boldsymbol{R}$ and Y, and determine which indicators are worth including, instead of adding all.

We propose a novel strategy to selectively add indicators based on a statistical test, which we call *Selective MIM* (SMIM). The overall procedure for SMIM is summarized in Algorithm 1. Instead of testing the relationship between R_j and $\boldsymbol{X}$, as a test for the missing mechanism would, SMIM tests the relationship between R_j and Y, directly yielding signal from the informative missingness towards the response of each missing indicator. Since we assume Y is complete, there are no issues testing the relationship between R_j and Y, as there would be when testing for the missing mechanism. Further, if X_j is correlated with Y, then testing the relationship between R_j and Y serves as a proxy for testing the relationship between X_j and R_j, which is otherwise impossible if X_j has missing values.

Specifically, for feature j, SMIM tests whether R_j and Y are independent. When Y is continuous, we use a two-sample t-test comparing the $E[Y \mid R = 1]$ and $E[Y \mid R = 0]$. When Y is discrete, we use a chi-squared test of independence between Y and R_j. To correct for multiple testing across the features, we use Benjamini-Hochberg p-value correction [3] with false discovery rate (FDR) of α. Since false negatives (not adding an indicator that is informative) are more harmful than false positives (adding an indicator that is not informative), we use a relatively high FDR of $\alpha = 0.1$ throughout the paper. See Algorithm 1.

5 EXPERIMENTS

We now empirically evaluate MIM and SMIM through experiments on synthetic data as well as real-world OpenML data sets with synthetic missing values. We show that MIM boosts performance when missing values are informative, and does not negatively affect performance on most data sets with uninformative missing values. Further, we show that SMIM can successfully discover the missing indicators that are informative, and is more effective than MIM on high-dimensional data sets.

5.1 Setup

For all experiments in this section, we start with complete data and mask values, controlling for how much the missing pattern is

Algorithm 1 Selective MIM

```
Input: Missing indicators $\boldsymbol{R}$, response $Y$, error rate $\alpha$.
Output: Indicator indices to keep $\mathcal{I} \subseteq \{1, \ldots, p\}$.
pvals ← [ ]
for $j = 1$ to $p$ do
    if $Y$ is continuous then
        pval ← t-test($Y \mid R = 0,\ Y \mid R = 1$)
    else if $Y$ is categorical then
        pval ← Chi2-test(Contingency-Table($Y, R_j$))
    end if
    pvals[$j$] ← pval
end for
reject = Benjamini–Hochberg(pvals, $\alpha$)
return $\mathcal{I} = \{j : \text{reject}_j = \text{true}\}$.
```

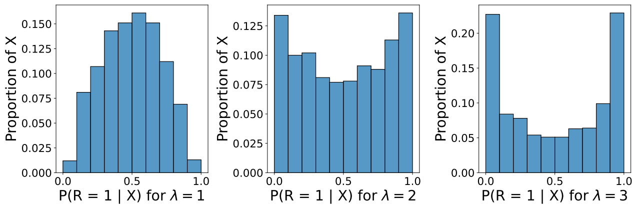

Figure 1: Distributions of masking probabilities $P(R = 1 \mid X)$ generated from $X \sim \mathcal{N}(0, 1)$ using Eq. (17) for different values of the informativeness parameter λ. Larger values of λ result in 'steeper' sigmoid functions in Eq. (17), hence more values lie in the masked-with-high-probability regime.

informative. To generate the missing masks, we use a self-masking mechanism:

$$P(R_j = 0 \mid \boldsymbol{X}) = P(R_j = 0 \mid X_j) = \frac{1}{1 + \exp(-\lambda_j X_j)}. \tag{17}$$

We call λ_j the *informativeness parameter* as it directly controls how informative the missing values in X_j are by determining the steepness of the sigmoid masking probability function. The higher the value of λ_j, the more informative the missing values are. If $\lambda_j = 0$ for all j, then the missing values are MCAR and the missingness is uninformative. The impact of λ_j on R_j is showcased in more detail in Figure 1.

We consider the following preprocessing methods to treat missing values:

- **Mean:** Mean imputation over the observed entries. This is equivalent to 0-imputation, since we always standardize features.
- **MF:** Imputation via missForest [45], an iterative imputer based on random forests [5].
- **GC:** Imputation via gcimpute [56, 57], an EM-style imputer which uses the Gaussian Copula.
- **LRGC:** Imputation via gcimpute with the low rank Gaussian Copula model [55], useful for high-dimensional data sets.
- **Imputer + MIM:** Imputation via the given imputer, along with MIM.

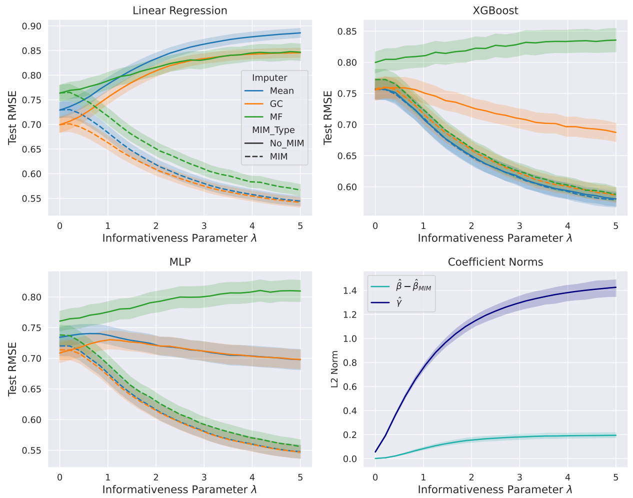

Figure 2: MIM performance on synthetic data with $(n, p) = (10000, 10)$ with different imputation methods, as a function of the informativeness parameter λ (see Eq. (17)). The top left, top right, and bottom left plots demonstrate that MIM (dashed lines) reduces test RSME in almost all scenarios when missingness is informative. The bottom right plot shows the linear regression coefficient norms, where $||\hat{\gamma}||$ increases as missing value become more informative, as shown in Theorem 3.2.

- **Imputer + SMIM:** Imputation via the given imputer, along with Selective MIM as in Algorithm 1.

We occasionally use the term "No MIM" to refer to imputation without MIM or SMIM. We choose these imputation methods to include either an iterative approach (MF) and an EM-based approach (GC), both popular classes of imputation methods. For supervised learning models, we use linear/logistic regression, XGBoost [7], and a multi-layer perceptron (MLP), giving us one model from the 3 most popular classes of supervised learning models (linear models, boosted decision trees, and neural networks). We standardize features over the observed entries in all experiments. We use a 75%-25% train-test data split and run each experiment for 20 trials, each with a different model seed, train-test split, and missing value mask. To isolate the impact of the missing value handling, we analyze how different missing value preprocessing methods impact each supervised learning method separately, rather than comparing the supervised learning methods to each other. For performance metrics, we use RMSE for regression tasks, 1 - AUC for binary classification tasks, and 1 - accuracy for multiclass classification tasks (so in all cases *lower is better*). We release the code necessary to reproduce our results [1].

5.2 MIM on Tree-Based Models

Before discussing our empirical results, we preface our findings with intuition for why XGBoost will likely behave differently with MIM than linear models and MLPs. Tree-based methods treat all

[1] https://github.com/mvanness354/missing_indicator_method.

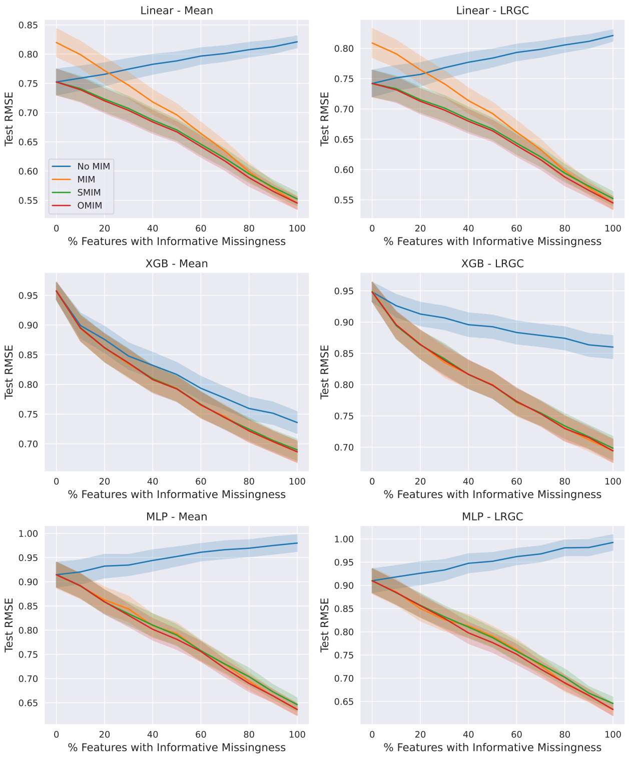

Figure 3: Comparison of MIM, SMIM, Oracle MIM (OMIM), and No MIM on high-dimensional synthetic data with $(n, p) = (10000, 1000)$, as a function of the percent of features with informative missingness.

features as discrete, and so they discretize continuous features using binning [6]. When using a constant imputation method like mean imputation, all imputed values always fall into the same bin and will always be split together by each tree. We thus expect MIM to provide little additional information to the tree-based methods in this setting, since splitting on the indicator feature can be alternatively achieved by splitting twice to isolate the bin with the constant imputation value. Meanwhile, if a non-constant imputation method is used, then MIM does add new splits for the tree to search over and thus has high potential to be valuable. On the other hand, linear and neural network models do not treat continuous features as discrete, and thus are expected to benefit from missing indicators with all imputation methods. A more detailed discussion of missing values in decision trees can be found in Appendix B.1.

5.3 Synthetic Data

Low Dimensional Data. We first consider the effects of MIM on synthetic data as a function of the informativeness parameter λ. Using $n = 10000$ and $p = 10$, we generate $\boldsymbol{X} \sim \mathcal{N}(0, \Sigma)$ with $\Sigma = \rho\mathbf{1}\mathbf{1}^T + (1-\rho)\mathcal{I}$ using $\rho = 0.3$ and $Y = \boldsymbol{X}^T\boldsymbol{\beta} + \epsilon$ for $\boldsymbol{\beta} \sim \mathcal{N}(0, \mathcal{I})$ and $\epsilon \sim \mathcal{N}(0, \sigma^2)$ with σ^2 chosen to enforce a signal-to-noise ratio of 10. We mask each feature according to Eq. (17) with $\lambda_j = \lambda$ for all $j = 1, \ldots, p$. The results for $\lambda \in [0, 5]$ are shown in Figure 2. The top left, top right, and bottom left plots show that MIM continually reduces RSME as λ increases across all imputation methods, which confirms that MIM is an effective preprocessor for informative missing values. The lone exception is XGBoost, which benefits from MIM using GC and MF imputation but not when using mean imputatation. This might be because mean imputation already allows trees to capture the informative signal in the missing values without MIM, as explained in Section 5.2.

The bottom right plot shows the linear regression coefficient norms $||\hat{\boldsymbol{\beta}} - \hat{\boldsymbol{\beta}}_{MIM}||_2$ and $||\hat{\boldsymbol{\gamma}}||_2$ as a function of λ, using the definition of these coefficients from Section 3. When $\lambda = 0$, both norms are close to 0, which we expect from Theorem 3.2 (a) as it tells us that $\hat{\boldsymbol{\beta}} \approx \hat{\boldsymbol{\beta}}_{MIM}$ and $\hat{\boldsymbol{\gamma}} \approx 0$ when missingness is uninformative. As λ increases, i.e. the missing values become more informative, $||\hat{\boldsymbol{\gamma}}||_2$ increases while $||\hat{\boldsymbol{\beta}} - \hat{\boldsymbol{\beta}}_{MIM}||_2$ remains small. This behavior parallels Theorem 3.2 (b), which shows that under a self-masking mechanism like Eq. (17), $||\hat{\boldsymbol{\beta}} - \hat{\boldsymbol{\beta}}_{MIM}||_2$ should remain small but $||\hat{\boldsymbol{\gamma}}||_2$ should increase to capture the informative signal in each variable, i.e. $\mathbb{E}[Y \mid R_j = 1] - \mathbb{E}[Y \mid R_j = 0]$. It is noteworthy that this behavior still holds even without independence among the features in $\boldsymbol{X}$, suggesting that this behavior might hold, at least approximately, in more general linear regression settings.

High Dimensional Data. We now consider high-dimensional synthetic data with $n = 10000$ and $p = 1000$. To generate realistic high-dimensional data, we generate $\boldsymbol{X} \sim \mathcal{N}(0, \Sigma)$ where Σ is block diagonal with b blocks and d features per block, with block elements $\rho\mathbf{1}\mathbf{1}^T + (1-\rho)\mathcal{I}$ using $\rho = 0.5$. We compute $\boldsymbol{X}^* \in \mathbb{R}^b$ as the block-wise mean of $\boldsymbol{X}$ (averaging over features in each block), and generate Y as in the low dimensional case: $Y = \boldsymbol{X}^{*T}\boldsymbol{\beta} + \epsilon$ with $\epsilon \sim \mathcal{N}(0, \sigma^2)$. This simulates real-world high-dimensional data where several features may be correlated with each other (but predominantly independent of other features), and thus are more likely to be missing together under informative missingness.

To generate the mask, we select only a random subset of blocks to have informative missingness: for each block, with probability p_{inf} we set $\lambda = 2$ for all features in the block as a whole, and with probability $1 - p_{\text{inf}}$ we set $\lambda = 0$ for all features in said block. In this setting, SMIM should be able to detect which features have informative missing values and only add the correponding indicators. Along with MIM and SMIM, we also run experiments with Oracle MIM (OMIM), in which only features masked with $\lambda_j = 2$ are added, to represent an unrealistic but optimal preprocessor.

The results for $p_{\text{inf}} \in [0, 1]$ are shown in Figure 3. We use LRGC instead of GC since the data is high-dimensional, and we do not report MF because computations do not terminate in less than 3 hours. Using linear models, SMIM achieves comparable RMSE to OMIM and significantly better RMSE than No MIM across all values of p_{inf}. Further, SMIM outperforms MIM for smaller values of p_{inf}, when MIM adds many uninformative indicators. For MLP models, MIM and SMIM achieve comparable error to OMIM across all p_{inf}, demonstrating that MLPs can more readily ignore uninformative feature than linear models. Lastly, XGBoost with mean imputation is comparable for preprocessors, but suffers from No MIM with LRGC, further supporting the discussion on tree-based models in Section 5.2.

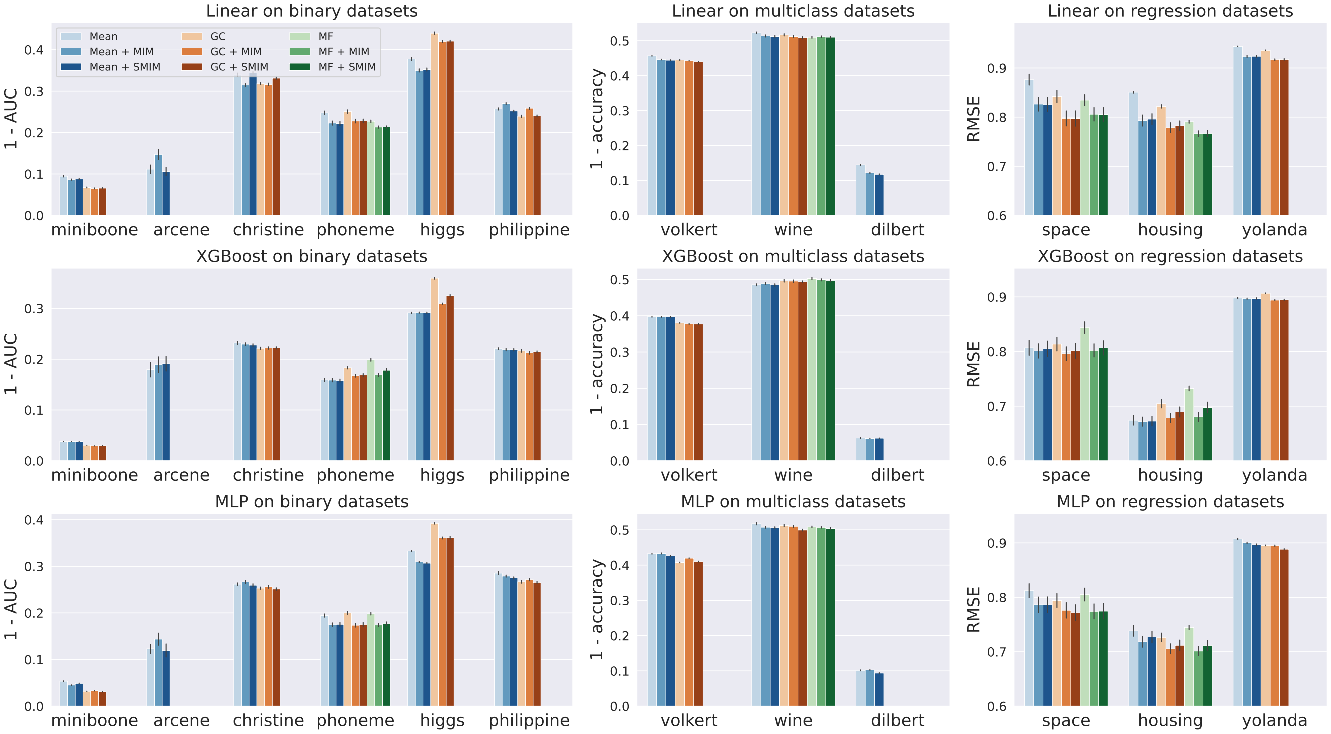

Figure 4: Test performance (lower is better) of MIM and SMIM with various imputation methods on OpenML data sets. Missing values are generated according to Eq. (17), where for each feature $\lambda_j = 2$ with probability $p_{\mathbf{inf}} = 0.5$ and $\lambda_j = 0$ otherwise. Performance metric is RMSE for regression problems, 1-AUC for binary classification problems, 1-accuracy for multi-class problems. Each bar represents the mean across 20 trials, and a a missing bar indicates a >3 hour run time.

5.4 Masked Real-World Data

We now run experiments on fully-observed real-world data sets obtained from the OpenML data set repository [50]. We mask entries according to Eq. (17), using $\lambda_j = 2$ with probability 0.5 and $\lambda_j = 0$ with probability 0.5 for all features j. We select 12 data sets that cover a diverse spectrum of values for n (number of samples), p (number of features), and outcome type (binary, multiclass, regression); see Appendix C.1 for further OpenML data set descriptions. We focus on data sets with continuous features, although we discuss how MIM can be used with categorical features in Appendix B.2. Lastly, we use LRGC for high-dimensional data sets (arcene, christine, philippine, volkert, dilbert, yolanda) and use GC for all other data sets.

Figure 4 shows the performance of each missing value preprocessing method paired with the 3 supervised learning models. For Linear and MLP models, MIM and SMIM improve performance across imputation methods for almost all data sets. This affirms the importance of MIM when missing values are informative. XGBoost generally does as well with mean imputation as with other imputation methods, and is less impacted by MIM, supporting the discussion from Section 5.2. On high-dimensional data sets, SMIM outperforms MIM in most cases. This further demonstrates the value of discarding uninformative features in high-dimensional data, which SMIM can do effectively.

To better understand the time efficiency of the methods employed, we plot the relative wallclock times of each result from Figure 4 in Figure 5. It is clear from Figure 5 that the choice of imputation method dominates the computation time, instead of whether or not MIM or SMIM is used. Specifically, mean imputation is orders of magnitude faster than GC and MF imputation. Also, MIM and SMIM appear to add relatively very little time to the total computation time, even though MIM doubles the number of features, suggesting that the method of imputation is much more important than the inclusion of MIM. Further, notice that in Figure 4, (S)MIM with mean imputation often performs very comparably to (S)MIM with GC and MF. Therefore, (S)MIM with mean imputation is an effective yet remarkably efficient alternative to using expensive imputation models.

6 MIM ON THE MIMIC BENCHMARK

In Section 5, we demonstrate that MIM and SMIM are effective tools for capturing informative missingness in synthetic and real-world data sets with synthetic missing values. In this section, we demonstrate the effectiveness of MIM and SMIM on healthcare data which already has missing values. Specifically, we use MIMIC-III [20], an open source clinical database of electronic health records (EHRs)

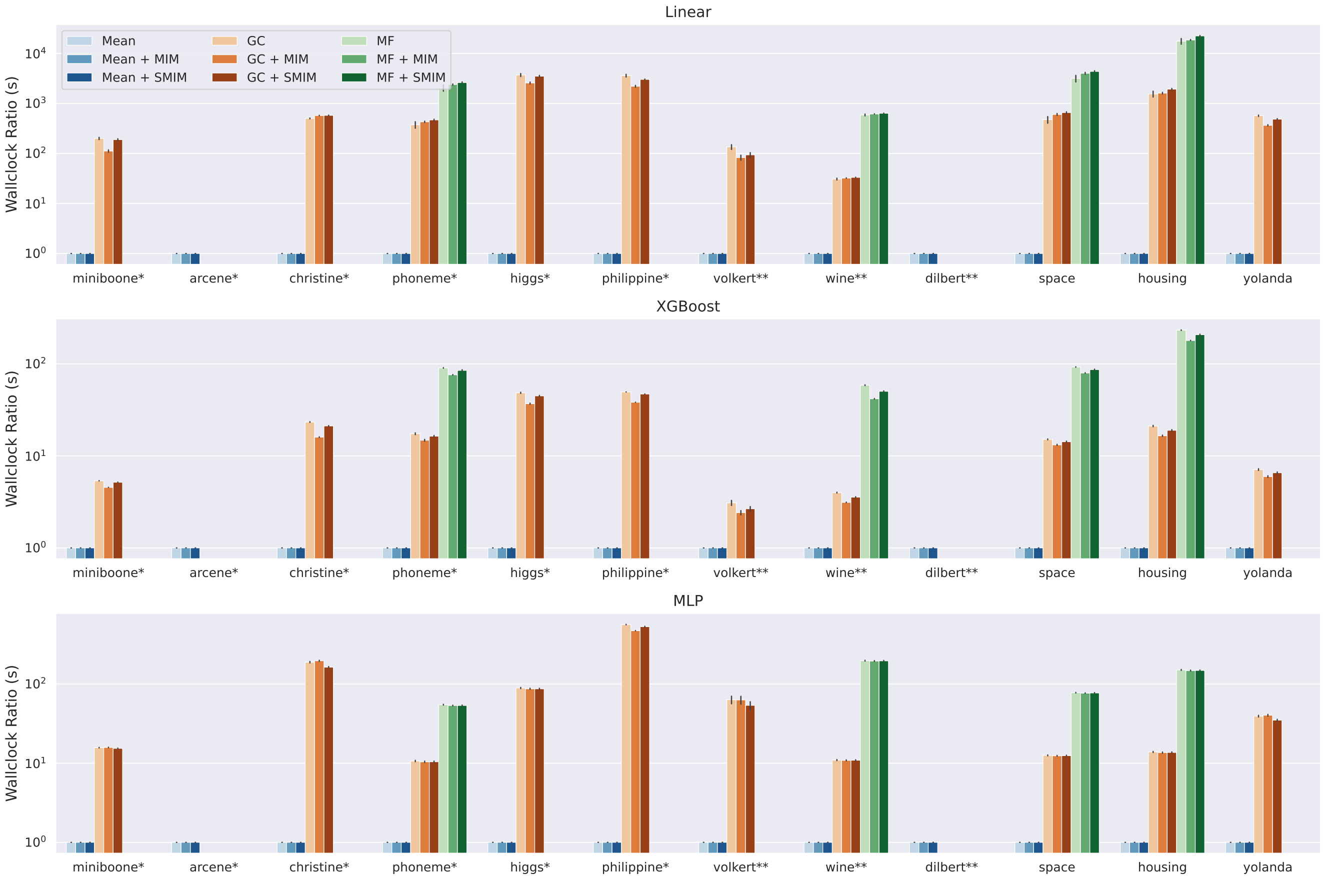

Figure 5: Wallclock times (in seconds) for results in Figure 4. Each time is the ratio between the indicated method and the corresponding mean imputation method. A missing bar indicates a >3 hour run time, as in Figure 4.

from the Beth Israel Deaconess Medical Center in Boston, Massachusetts. To preprocess the MIMIC data, we utilize the mimic3-benchmark [17], which generates tabular data for various clinical tasks. We consider the following tasks for our experiments:

- **In Hospital Mortality Prediction**: binary mortality prediction using the first 48 hours of an ICU stay.
- **Length-of-Stay Prediction (LOS)**: prediction of the remaining length-of-stay in the hospital. We formulate this as a binary prediction task to predict whether or not a patient will leave the hospital in the next 7 days.
- **Phenotyping**: prediction of the acute care conditions present during a given hospital stay, formulated as a multi-label binary classification problem for 25 phenotypes.

For each task, the target metric is 1 - AUC, following Section 5 and using a metric such that lower is better (for consistency across response types). For phenotyping, we average the AUC for each of the 25 labels to create the macro AUC score, then set the metric as 1 - macro AUC. The train and test splits are established by the benchmark and kept constant across all trials. The data set for each task contains 17 clinical variables that describe patient vital signs and other typical hospital measurements. For additional details about our these variables and well as our experimental setup for MIMIC experiments, see Appendix C.3.

Figure 6 shows the results of MIM and SMIM with the prediction pipelines used in Section 5. As on synthetic data and OpenML data, MIM and SMIM both consistently improve predictive performance with linear and MLP models, as well as with XGBoost when using GC or MF for imputations. This result is particularly significant because it demonstrates that real-world data sets often have informative missing values, and MIM can help predictive models learn the signal in these missing values. Additionally, there are no scenarios MIM negatively impacts predictive performance, confirming that MIM should be a standard preprocessing tool for supervised learning on low-dimensional real-world data sets, should predictive performance be prioritized. In Figure 7 in Appendix C.2, we provide additional evidence that real-world data often contains informative missing values through additional experiments on OpenML data sets.

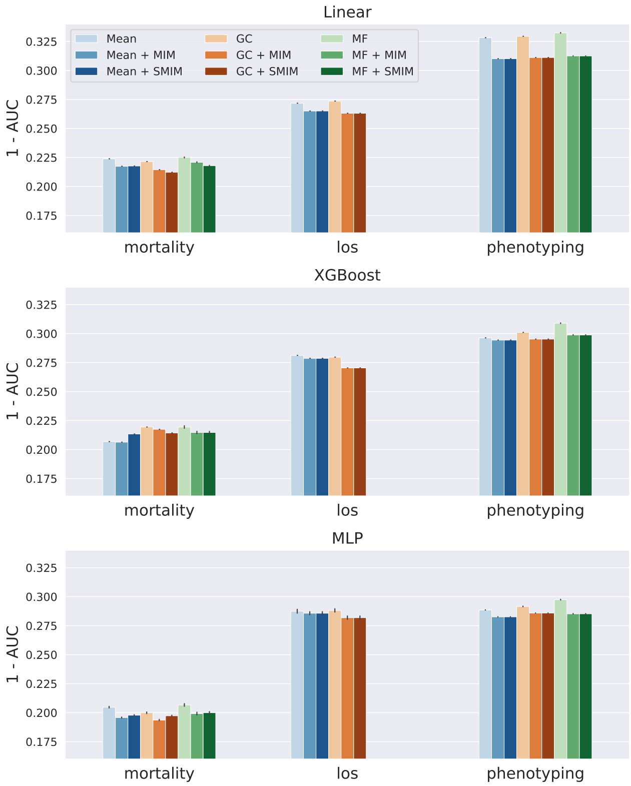

Figure 6: MIM and SMIM performance (lower is better) for clinical tasks on the MIMIC-III data set. Each bar represents the mean across 20 trials, each with a different random seed. A missing bar indicates a >3 hour run time. MIM and SMIM improve performance for logistic regression and MLP models on all 3 tasks, following the behavior exhibited in Figure 4.

7 DISCUSSION

From our experiments on synthetic data, real-world data with synthetic missing values, and real-world EHR data, we obtain the following main takeaways:

- When missing values are informative, MIM increases predictive performance of linear models and neural networks. This is supported by Figure 2, where increasing the informativeness in synthetic data gradually increases the effectiveness of MIM; in OpenML data with informative missing values in Figure 4; and in EHR data in Figure 6.
- Tree-based models in general benefit less from MIM than linear models and neural networks, particularly when using mean imputation. We provide a possible explanation for this behavior in Section 5.2, and see this behavior manifest across all of our experiments.
- The only scenario where MIM might harm predictive performance is on high-dimensional data sets, e.g. the synthetic data in Figure 3 and on select high-dimensional OpenML data sets in Figure 4. In these cases, Selective MIM (SMIM) is a stable extension of MIM that adds all the indicator features that have informative missing values.
- MIM and SMIM can increase performance not only with mean imputation, but also with other imputation methods, as shown in our experiments. Nonetheless, Figure 5 shows that MIM with mean imputation is many orders of magnitude faster than using other imputation methods, yet usually results in about the same predictive performance. Therefore, MIM with mean imputation is a very effective and yet efficient way to treat missing values, especially under informative missingness when expensive imputation methods often bring little benefit.

8 CONCLUSION

When dealing with missing data, imputation with state-of-the-art imputers is often expensive. We show that using MIM in conjunction with mean imputation is an effective and efficient alternative, which we demonstrate via novel theory and comprehensive experimentation on synthetic data as well as real data with both synthetic and actual missing values. We additionally introduce Selective MIM (SMIM), a MIM-based preprocessor that discards uninformative missing indicators and is thus more stable than MIM on high-dimensional data sets. We show experimentally that adding MIM or SMIM helps achieve substantially better accuracy on data with informative missingness overall, and SMIM outperforms MIM on high-dimensional data. Future work might include researching theoretical guarantees on the use of imputation methods along with missing indicators, building on our theory.

REFERENCES

[1] Amanda N Baraldi and Craig K Enders. 2010. An introduction to modern missing data analyses. *Journal of school psychology* 48, 1 (2010), 5–37.

[2] Brett K Beaulieu-Jones, Patryk Orzechowski, and Jason H Moore. 2018. Mapping patient trajectories using longitudinal extraction and deep learning in the MIMIC-III critical care database. In *PACIFIC SYMPOSIUM ON BIOCOMPUTING 2018: Proceedings of the Pacific Symposium*. World Scientific, 123–132.

[3] Yoav Benjamini and Yosef Hochberg. 1995. Controlling the false discovery rate: a practical and powerful approach to multiple testing. *Journal of the Royal statistical society: series B (Methodological)* 57, 1 (1995), 289–300.

[4] Dimitris Bertsimas, Arthur Delarue, and Jean Pauphilet. 2021. Prediction with Missing Data. *arXiv preprint arXiv:2104.03158* (2021).

[5] Leo Breiman. 2001. Random forests. *Machine learning* 45 (2001), 5–32.

[6] Leo Breiman. 2017. *Classification and regression trees*. Routledge.

[7] Tianqi Chen and Carlos Guestrin. 2016. Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*. 785–794.

[8] Arthur P Dempster, Nan M Laird, and Donald B Rubin. 1977. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society: Series B (Methodological)* 39, 1 (1977), 1–22.

[9] Ning Ding, Cuirong Guo, Changluo Li, Yang Zhou, and Xiangping Chai. 2021. An artificial neural networks model for early predicting in-hospital mortality in acute pancreatitis in MIMIC-III. *BioMed research international* 2021 (2021).

[10] Matthias Feurer, Katharina Eggensperger, Stefan Falkner, Marius Lindauer, and Frank Hutter. 2020. Auto-Sklearn 2.0: Hands-free automl via meta-learning. *arXiv preprint arXiv:2007.04074* (2020).

[11] Unai Garciarena and Roberto Santana. 2017. An extensive analysis of the interaction between missing data types, imputation methods, and supervised classifiers. *Expert Systems with Applications* 89 (2017), 52–65.

[12] Unai Garciarena, Roberto Santana, and Alexander Mendiburu. 2017. Evolving imputation strategies for missing data in classification problems with TPOT. *arXiv preprint arXiv:1706.01120* (2017).

[13] Thanos Gentimis, Alnaser Ala'J, Alex Durante, Kyle Cook, and Robert Steele. 2017. Predicting hospital length of stay using neural networks on mimic iii data. In *2017 IEEE 15th intl conf on dependable, autonomic and secure computing, 15th intl conf on pervasive intelligence and computing, 3rd intl conf on big data intelligence and computing and cyber science and technology congress (DASC/PiCom/DataCom/CyberSciTech)*. IEEE, 1194–1201.

[14] Lovedeep Gondara and Ke Wang. 2018. Mida: Multiple imputation using denoising autoencoders. In *Advances in Knowledge Discovery and Data Mining: 22nd Pacific-Asia Conference, PAKDD 2018, Melbourne, VIC, Australia, June 3-6, 2018, Proceedings, Part III 22*. Springer, 260–272.

[15] Rolf HH Groenwold, Ian R White, A Rogier T Donders, James R Carpenter, Douglas G Altman, and Karel GM Moons. 2012. Missing covariate data in clinical research: when and when not to use the missing-indicator method for analysis. *Cmaj* 184, 11 (2012), 1265–1269.

[16] Bruce Hansen. 2022. *Econometrics*. Princeton University Press.

[17] Hrayr Harutyunyan, Hrant Khachatrian, David C. Kale, Greg Ver Steeg, and Aram Galstyan. 2019. Multitask learning and benchmarking with clinical time series data. *Scientific Data* 6, 1 (2019), 96. https://doi.org/10.1038/s41597-019-0103-9

[18] James Honaker, Gary King, and Matthew Blackwell. 2011. Amelia II: A program for missing data. *Journal of statistical software* 45 (2011), 1–47.

[19] Niels Bruun Ipsen, Pierre-Alexandre Mattei, and Jes Frellsen. 2021. How to deal with missing data in supervised deep learning?. In *International Conference on Learning Representations*.

[20] Alistair EW Johnson, Tom J Pollard, Lu Shen, Li-wei H Lehman, Mengling Feng, Mohammad Ghassemi, Benjamin Moody, Peter Szolovits, Leo Anthony Celi, and Roger G Mark. 2016. MIMIC-III, a freely accessible critical care database. *Scientific data* 3, 1 (2016), 1–9.

[21] Julie Josse, Nicolas Prost, Erwan Scornet, and Gaël Varoquaux. 2019. On the consistency of supervised learning with missing values. *arXiv preprint arXiv:1902.06931* (2019).

[22] Mirjam J Knol, Kristel JM Janssen, A Rogier T Donders, Antoine CG Egberts, E Rob Heerdink, Diederick E Grobbee, Karel GM Moons, and Mirjam I Geerlings. 2010. Unpredictable bias when using the missing indicator method or complete case analysis for missing confounder values: an empirical example. *Journal of clinical epidemiology* 63, 7 (2010), 728–736.

[23] Trent Kyono, Yao Zhang, Alexis Bellot, and Mihaela van der Schaar. 2021. MIRACLE: Causally-Aware Imputation via Learning Missing Data Mechanisms. *Advances in Neural Information Processing Systems* 34 (2021).

[24] Marine Le Morvan, Julie Josse, Thomas Moreau, Erwan Scornet, and Gaël Varoquaux. 2020. NeuMiss networks: differentiable programming for supervised learning with missing values. *Advances in Neural Information Processing Systems* 33 (2020), 5980–5990.

[25] Marine Le Morvan, Julie Josse, Erwan Scornet, and Gaël Varoquaux. 2021. What's a good imputation to predict with missing values? *Advances in Neural Information Processing Systems* 34 (2021).

[26] Marine Le Morvan, Nicolas Prost, Julie Josse, Erwan Scornet, and Gaël Varoquaux. 2020. Linear predictor on linearly-generated data with missing values: non consistency and solutions. In *International Conference on Artificial Intelligence and Statistics*. PMLR, 3165–3174.

[27] Jia Li, Mengdie Wang, Michael S Steinbach, Vipin Kumar, and Gyorgy J Simon. 2018. Don't do imputation: Dealing with informative missing values in EHR data analysis. In *2018 IEEE International Conference on Big Knowledge (ICBK)*. IEEE, 415–422.

[28] Wei-Chao Lin and Chih-Fong Tsai. 2020. Missing value imputation: a review and analysis of the literature (2006–2017). *Artificial Intelligence Review* 53, 2 (2020), 1487–1509.

[29] Roderick JA Little. 1988. A test of missing completely at random for multivariate data with missing values. *Journal of the American statistical Association* 83, 404 (1988), 1198–1202.

[30] Roderick JA Little and Donald B Rubin. 2019. *Statistical analysis with missing data*. Vol. 793. John Wiley & Sons.

[31] Pierre-Alexandre Mattei and Jes Frellsen. 2019. MIWAE: Deep generative modelling and imputation of incomplete data sets. In *International conference on machine learning*. PMLR, 4413–4423.

[32] Siddhartha Nuthakki, Sunil Neela, Judy W Gichoya, and Saptarshi Purkayastha. 2019. Natural language processing of MIMIC-III clinical notes for identifying diagnosis and procedures with neural networks. *arXiv preprint arXiv:1912.12397* (2019).

[33] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. 2011. Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research* 12 (2011), 2825–2830.

[34] Alexandre Perez-Lebel, Gaël Varoquaux, Marine Le Morvan, Julie Josse, and Jean-Baptiste Poline. 2022. Benchmarking missing-values approaches for predictive models on health databases. *GigaScience* 11 (2022).

[35] Sanjay Purushotham, Chuizheng Meng, Zhengping Che, and Yan Liu. 2018. Benchmarking deep learning models on large healthcare datasets. *Journal of biomedical informatics* 83 (2018), 112–134.

[36] Yongming Qu and Ilya Lipkovich. 2009. Propensity score estimation with missing values using a multiple imputation missingness pattern (MIMP) approach. *Statistics in medicine* 28, 9 (2009), 1402–1414.

[37] J Ross Quinlan. 2014. *C4. 5: programs for machine learning*. Elsevier.

[38] Donald B Rubin. 1976. Inference and missing data. *Biometrika* 63, 3 (1976), 581–592.

[39] Joseph L Schafer and John W Graham. 2002. Missing data: our view of the state of the art. *Psychological methods* 7, 2 (2002), 147.

[40] Matthieu Scherpf, Felix Gräßer, Hagen Malberg, and Sebastian Zaunseder. 2019. Predicting sepsis with a recurrent neural network using the MIMIC III database. *Computers in biology and medicine* 113 (2019), 103395.

[41] Anis Sharafoddini, Joel A Dubin, David M Maslove, Joon Lee, et al. 2019. A new insight into missing data in intensive care unit patient profiles: observational study. *JMIR medical informatics* 7, 1 (2019), e11605.

[42] Matthew Sperrin and Glen P Martin. 2020. Multiple imputation with missing indicators as proxies for unmeasured variables: simulation study. *BMC medical research methodology* 20, 1 (2020), 1–11.

[43] Matthew Sperrin, Glen P Martin, Rose Sisk, and Niels Peek. 2020. Missing data should be handled differently for prediction than for description or causal explanation. *Journal of Clinical Epidemiology* 125 (2020), 183–187.

[44] Aude Sportisse, Claire Boyer, and Julie Josse. 2020. Estimation and imputation in probabilistic principal component analysis with missing not at random data. *Advances in Neural Information Processing Systems* 33 (2020), 7067–7077.

[45] Daniel J Stekhoven and Peter Bühlmann. 2012. MissForest—non-parametric missing value imputation for mixed-type data. *Bioinformatics* 28, 1 (2012), 112–118.

[46] Shengpu Tang, Parmida Davarmanesh, Yanmeng Song, Danai Koutra, Michael W Sjoding, and Jenna Wiens. 2020. Democratizing EHR analyses with FIDDLE: a flexible data-driven preprocessing pipeline for structured clinical data. *Journal of the American Medical Informatics Association* 27, 12 (2020), 1921–1934.

[47] Bheki ETH Twala, MC Jones, and David J Hand. 2008. Good methods for coping with missing data in decision trees. *Pattern Recognition Letters* 29, 7 (2008), 950–956.

[48] Stef Van Buuren. 2018. *Flexible imputation of missing data*. CRC press.

[49] Stef Van Buuren and Karin Groothuis-Oudshoorn. 2011. mice: Multivariate imputation by chained equations in R. *Journal of statistical software* 45 (2011), 1–67.

[50] Joaquin Vanschoren, Jan N Van Rijn, Bernd Bischl, and Luis Torgo. 2014. OpenML: networked science in machine learning. *ACM SIGKDD Explorations Newsletter* 15, 2 (2014), 49–60.

[51] Shirly Wang, Matthew BA McDermott, Geeticka Chauhan, Marzyeh Ghassemi, Michael C Hughes, and Tristan Naumann. 2020. Mimic-extract: A data extraction, preprocessing, and representation pipeline for mimic-iii. In *Proceedings of the ACM conference on health, inference, and learning*. 222–235.

[52] Katarzyna Woźnica and Przemysław Biecek. 2020. Does imputation matter? Benchmark for predictive models. *arXiv preprint arXiv:2007.02837* (2020).

[53] Chengrun Yang, Jicong Fan, Ziyang Wu, and Madeleine Udell. 2020. Automl pipeline selection: Efficiently navigating the combinatorial space. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 1446–1456.

[54] Jinsung Yoon, James Jordon, and Mihaela Schaar. 2018. Gain: Missing data imputation using generative adversarial nets. In *International conference on machine learning*. PMLR, 5689–5698.

[55] Yuxuan Zhao and Madeleine Udell. 2020. Matrix completion with quantified uncertainty through low rank gaussian copula. *Advances in Neural Information Processing Systems* 33 (2020), 20977–20988.

[56] Yuxuan Zhao and Madeleine Udell. 2020. Missing value imputation for mixed data via gaussian copula. In *Proceedings of the 26th ACM SIGKDD international conference on knowledge discovery & data mining*. 636–646.

[57] Yuxuan Zhao and Madeleine Udell. 2022. gcimpute: A Package for Missing Data Imputation. *arXiv preprint arXiv:2203.05089* (2022).

[58] Yibing Zhu, Jin Zhang, Guowei Wang, Renqi Yao, Chao Ren, Ge Chen, Xin Jin, Junyang Guo, Shi Liu, Hua Zheng, et al. 2021. Machine learning prediction models for mechanically ventilated patients: analyses of the MIMIC-III database. *Frontiers in medicine* 8 (2021), 662340.

A PROOFS

A.1 Proof of Theorem 3.1

Let $\boldsymbol{D} = [\tilde{Z}, R]^T$, then

$$\boldsymbol{DD}^T = \begin{bmatrix}\tilde{Z} \\ R\end{bmatrix}\begin{bmatrix}\tilde{Z} & R\end{bmatrix} = \begin{bmatrix}\tilde{Z}^2 & 0 \\ 0 & R\end{bmatrix}. \tag{18}$$

as $R^2 = R$ and $\tilde{Z}R = 0$ by zero imputation (Assumption 3.1). Thus the finite sample OLS estimates $\hat{\beta}$ and $\hat{\gamma}$ are

$$\begin{bmatrix}\hat{\beta}_{MIM} \\ \hat{\gamma}_{MIM}\end{bmatrix} = \mathbb{E}_n\left[\boldsymbol{DD}^T\right]^{-1}\mathbb{E}_n\left[\boldsymbol{D}Y\right] \tag{19}$$

$$= \begin{bmatrix}\mathbb{E}_n\left[\tilde{Z}^2\right]^{-1}\mathbb{E}_n[\tilde{Z}Y] \\ \mathbb{E}_n\left[R\right]^{-1}\mathbb{E}_n[RY]\end{bmatrix} \tag{20}$$

$$= \begin{bmatrix}\hat{\beta} \\ \mathbb{E}_n[Y \mid R = 1]\end{bmatrix} \tag{21}$$

The result for γ in (21) holds because

$$\mathbb{E}_n\left[R\right]^{-1}\mathbb{E}_n[RY] = \left(\frac{|\mathcal{M}|}{n}\right)^{-1}\frac{1}{n}\sum_{i \in \mathcal{M}} Y^{(i)} \tag{22}$$

$$= \frac{1}{|\mathcal{M}|}\sum_{i \in \mathcal{M}} Y^{(i)} \tag{23}$$

$$= \mathbb{E}_n[Y \mid R = 1] \tag{24}$$

□

A.2 Proof of Theorem 3.2

Part (a). Let $\boldsymbol{D} = [\tilde{\boldsymbol{Z}}^T, \boldsymbol{R}^T]^T$ and $p_j = P(R = 1)$. We would like to show that $\mathbb{E}\left[\boldsymbol{DD}^T\right]$ is a 2×2 block diagonal matrix. When $j \neq k$, $\mathbb{E}[\tilde{Z}_j R_k] = 0$ since X_j and R_k are independent under MCAR. If $j = k$, we cannot assume that $\tilde{Z}_j$ and R_j are independent since they are directly dependent by construction. Nonetheless, by the law of total expectations we have

$$\mathbb{E}[\tilde{Z}_j R_j] = \mathbb{E}[\tilde{Z}_j R_j \mid R_j = 1]p_j + \mathbb{E}[\tilde{Z}_j R_j \mid R_j = 0](1 - p_j) \tag{25}$$

$$= \mathbb{E}[\tilde{Z}_j \mid R_j = 1]p_j \tag{26}$$

$$= 0 \tag{27}$$

since $\tilde{\boldsymbol{Z}}$ is imputed with 0. We have now shown that $\mathbb{E}[\tilde{Z}_j R_k] = 0$ for all j, k, showing as desired that $\mathbb{E}\left[\boldsymbol{DD}^T\right]$ is block diagonal. Further, since R is uninformative, we have for all j

$$\mathbb{E}[R_j Y] = \mathbb{E}[R_j]\mathbb{E}[Y] = 0 \tag{28}$$

using the centering of Y. Thus we have

$$\begin{bmatrix}\boldsymbol{\beta}^*_{MIM} \\ \boldsymbol{\gamma}^*_{MIM}\end{bmatrix} = \left(\mathbb{E}[\boldsymbol{DD}^T]\right)^{-1}\mathbb{E}[\boldsymbol{D}Y] \tag{29}$$

$$= \begin{bmatrix}\mathbb{E}[\tilde{\boldsymbol{Z}}\tilde{\boldsymbol{Z}}^T]^{-1} & 0 \\ 0 & \mathbb{E}[\boldsymbol{RR}^T]^{-1}\end{bmatrix}\begin{bmatrix}\mathbb{E}[\tilde{\boldsymbol{Z}}Y] \\ 0\end{bmatrix} \tag{30}$$

$$= \begin{bmatrix}\boldsymbol{\beta}^* \\ 0\end{bmatrix}, \tag{31}$$

□

Part (b). For Part b we assume that $\boldsymbol{R}$ is centered, so

$$R_j = \begin{cases}1 - p_j & \text{w.p. } p_j \\ -p_j & \text{w.p. } 1 - p_j\end{cases}. \tag{32}$$

Like in the proof of (a), we want to show that $\mathbb{E}[\boldsymbol{DD}^T]$ is 2×2 is 2×2 block diagonal. The only difference from the proof used in part (a) is that $\boldsymbol{R}$ is now centered. $\mathbb{E}[\tilde{Z}_j R_k] = 0$ when $j \neq k$ because X_j and R_k are still independent under the self-masking mechanism, and when $j = k$,

$$\mathbb{E}[\tilde{Z}_j R_j] = \mathbb{E}[\tilde{Z}_j R_j \mid R_j = 1 - p_j]p_j + \mathbb{E}[\tilde{Z}_j R_j \mid R_j = -p_j](1 - p_j) \tag{33}$$

$$= p_j(1 - p_j)\left(\mathbb{E}[\tilde{Z}_j \mid R_j = 1 - p_j] - \mathbb{E}[\tilde{Z}_j \mid R_j = -p_j]\right) \tag{34}$$

$$= 0 \tag{35}$$

where $\mathbb{E}[\tilde{Z}_j \mid R_j = 1 - p_j] = 0$ because of 0 imputation and $\mathbb{E}[\tilde{Z}_j \mid R_j = -p_j] = 0$ because of centering.

Different from (a), though, $R_j \not\perp\!\!\!\perp Y$ now because both depend on X_j. We thus have

$$\mathbb{E}[R_j Y] = \mathbb{E}[YR_j \mid R_j = 1 - p_j]p_j + \mathbb{E}[YR_j \mid R_j = -p_j](1 - p_j) \tag{36}$$

$$= p_j(1 - p_j)\left(\mathbb{E}[Y \mid R_j = 1 - p_j] - \mathbb{E}[Y \mid R_j = -p_j]\right). \tag{37}$$

Lastly, $\mathbb{E}[R_j, R_k] = 0$ when $j = k$ since $R_j \perp\!\!\!\perp R_k$ under self-masking, and $\mathbb{E}[R_j^2] = p_j(1 - p_j)$. Putting this together, we have

$$\begin{bmatrix}\boldsymbol{\beta}^*_{MIM} \\ \boldsymbol{\gamma}^*_{MIM}\end{bmatrix} = \begin{bmatrix}\mathbb{E}[\tilde{\boldsymbol{Z}}\tilde{\boldsymbol{Z}}^T]^{-1} & 0 \\ 0 & \mathbb{E}[\boldsymbol{RR}^T]^{-1}\end{bmatrix}\begin{bmatrix}\mathbb{E}[\tilde{\boldsymbol{Z}}Y] \\ \mathbb{E}[\boldsymbol{R}Y]\end{bmatrix} \tag{38}$$

$$= \begin{bmatrix}\beta^*_1 \\ \vdots \\ \beta^*_p \\ \mathbb{E}[Y \mid R_1 = 1 - p_1] - \mathbb{E}[Y \mid R_1 = -p_1] \\ \vdots \\ \mathbb{E}[Y \mid R_p = 1 - p_p] - \mathbb{E}[Y \mid R_p = -p_p]\end{bmatrix}. \tag{39}$$

□

Part (c). The proof of part (c) starts by reordering the columns of $\boldsymbol{D} = [\tilde{\boldsymbol{Z}}^T, \boldsymbol{R}^T]^T$ based on the blocks $B_1, \ldots, B_d$, i.e.

$$\boldsymbol{D} := [\boldsymbol{D}_{B_1}^T, \ldots, \boldsymbol{D}_{B_d}^T]^T := [\tilde{\boldsymbol{Z}}_{B_1}^T, \boldsymbol{R}_{B_1}^T, \ldots, \tilde{\boldsymbol{Z}}_{B_d}^T, \boldsymbol{R}_{B_d}^T]^T. \quad (40)$$

Using the same logic as in the proof of part (b), we can conclude that $\mathbb{E}\left[\boldsymbol{D}\boldsymbol{D}^T\right]$ is $d \times d$ block diagonal, where d is the number of blocks, and further for block B_ℓ

$$\begin{bmatrix} \boldsymbol{\beta}^*_{MIM_{B_\ell}} \\ \boldsymbol{\gamma}^*_{MIM_{B_\ell}} \end{bmatrix} = \mathbb{E}[\boldsymbol{D}_{B_\ell}\boldsymbol{D}_{B_\ell}^T]^{-1}\mathbb{E}[\boldsymbol{D}_{B_\ell}Y]. \quad (41)$$

Recall from the proof of (b) that

$$\mathbb{E}[R_j Y] = p_j(1-p_j)\left(\mathbb{E}[Y \mid R_j = 1 - p_j] - \mathbb{E}[Y \mid R_j = -p_j]\right) \quad (42)$$

thus for block B_ℓ

$$\mathbb{E}[\boldsymbol{D}_{B_\ell}\boldsymbol{D}_{B_\ell}^T]^{-1}\mathbb{E}[\boldsymbol{D}_{B_\ell}Y] \quad (43)$$

$$= \mathbb{E}[\boldsymbol{D}_{B_\ell}\boldsymbol{D}_{B_\ell}^T]^{-1}\begin{bmatrix} \mathbb{E}[\tilde{\boldsymbol{Z}}_{B_\ell}Y] \\ \mathbb{E}[Y \mid \boldsymbol{R}_{B_\ell} = 1 - \boldsymbol{p}_{B_\ell}] - \mathbb{E}[Y \mid \boldsymbol{R}_{B_\ell} = -\boldsymbol{p}_{B_\ell}] \end{bmatrix} \quad (44)$$

where we've abused notation and used $\mathbb{E}[Y \mid \boldsymbol{R}_{B_\ell} = 1 - \boldsymbol{p}_{B_\ell}]$ to mean the vector of $\mathbb{E}[Y \mid R_j = 1 - p_j]$ for $j \in B_\ell$ and similarly for $\mathbb{E}[Y \mid \boldsymbol{R}_{B_\ell} = -\boldsymbol{p}_{B_\ell}]$. □

B ADDITIONAL MIM DETAILS

B.1 MIM with decision trees

Supervised learning models based on decision trees are discrete models, making them different from continuous supervised learning models, like linear models and neural networks. When decision trees split on categorical features, the split is chosen based on the levels of the categorical feature. With numerical features, however, decision trees must choose a collection of threshold points. For each threshold point T, the tree can split on the numerical feature if it is greater than T or less than T. Exact details of how decision tree algorithms handle numerical features depends on the implementation; see, for example, the popular C4.5 decision tree algorithm [37].

When doing imputation as a preprocessing step for tree-based models like XGBoost, the choice of imputation can significantly affect the performance of the model. In particular, if the imputation method is constant, i.e. always imputes the same value, then the branches will be split by these imputed values each tree with high probability. With an imputation method that is non-constant, e.g. GC or MF, this is no longer the case. Additionally, for non-decision-tree-based supervised models, constant and non-constant imputation can have similar behavior in some cases, e.g. using constant imputation plus noise.

Now consider using MIM in conjunction with imputation for tree-based models. Each new indicator feature gives the tree 1 extra split to consider. Specifically, the indicator features allow a tree to split directly on whether or not a feature is observed. If constant imputation is used, e.g. mean imputation, these indicator features add little additional value to the model, since all imputed values are already split together by the tree. On the other hand, if a non-constant imputation is used, e.g. GC or MF, then the tree has much higher probability to split on the missing indicators, which can potentially really enhance the model. This explains why, throughout the experiments, MIM improves the performance of GC and MF more than the performance of Mean for experiments with XGBoost.

Since decision trees are discrete models, there are also other preprocessing methods unique to trees for dealing with missing values. One notable example is the Missing Incorporate as Attribute (MIA) strategy [21, 34, 47], which is similar in spirit to MIM. To illustrate MIA, let x be a numerical feature to split, with missing values represented as $*$. For a given threshold T, a normal decision tree would consider the split $x \geq T$ versus $x < T$. MIA instead considers the following 3 splits for each split:

- ($x \leq T$ or $x = *$) versus $x > T$
- $x \leq T$ versus ($x > T$ or $x = *$)
- $x = *$ versus $x \neq *$

MIA allows trees to split on missing values while doing any imputation, thereby allowing tree to utilize informative signal from the missing values if such signal is present. Note that the third split in the above list is the same as splitting on indicator features from MIM. The first and second splits in the list essentially compute the optimal constant imputation value for each feature, rather than always using the same value, e.g. the mean. Thus, MIA is very similar to Mean + MIM, although using another non-constant imputer with MIM is still quite different than MIA. A further discussion of MIA and other methods for handling missing values in decision trees can be found in [21].

B.2 MIM with categorical features

In our experiments, we consider data sets only with numeric variables, for several reasons. First, many imputation methods cannot handle categorical variables (e.g. gcimpute), making comparisons more difficult. Second, mean imputation is not possible with categorical variables. The closest comparison is perhaps mode imputation, i.e. imputing with the most frequent category, but this has different properties than mean imputation and complicates the analysis. On the other hand, MIM has a very straightforward implementation for categorical variables: replace missing categorical values with a new "missing" category. After one-hot encoding, this corresponds exactly to adding a new indicator column as in the numeric case. For neural network models, this transformation could also correspond to learning a new embedding for the missing category.

C ADDITIONAL EXPERIMENT DETAILS

C.1 OpenML Data Sets

In Table 2 we show a description of the OpenML data sets used. The source data sets can easily by found by searching the IDs on OpenML. We chose these data sets to represent diversity in n, p, and outcome type. We also restrict to data sets with continuous features, as explained in Appendix B.2. For Figure 4, we only use data sets which have no missing values, since we want to be able to completely control the missing mechanism in the data. The listed values for p are the number of features used in the data sets for our experiments, which in some cases is different than the listed number of features on OpenML. For example, there are several features in the volkert data set which all entirely 0, and so we remove these features.

Table 1: Details of the 17 clinical features used for MIMIC experiments in Section 6. For each feature and each task, the missing rate is given, along with whether the missingness is informative or not, based on the SMIM procedure in Algorithm 1. Multiple features for each task have informative missingness, suggesting that MIM will likely improve model performance, which turns out to be the case in Figure 6.

	Mortality		LOS		Phenotyping	
Feature	Missing Rate	Is Informative	Missing Rate	Is Informative	Missing Rate	Is Informative
Capillary refill rate	0.984	no	0.973	yes	0.985	yes
Diastolic blood pressure	0.012	no	0.069	yes	0.055	yes
Fraction inspired oxygen	0.710	yes	0.698	yes	0.753	yes
Glascow coma scale eye opening	0.010	no	0.070	yes	0.055	yes
Glascow coma scale motor response	0.010	no	0.070	yes	0.056	yes
Glascow coma scale total	0.422	no	0.438	yes	0.477	yes
Glascow coma scale verbal response	0.010	no	0.070	yes	0.056	yes
Glucose	0.001	no	0.007	yes	0.011	yes
Heart Rate	0.012	no	0.068	yes	0.054	yes
Height	0.829	yes	0.805	yes	0.797	yes
Mean blood pressure	0.012	no	0.069	yes	0.056	yes
Oxygen saturation	0.006	no	0.059	yes	0.048	yes
Respiratory rate	0.012	no	0.068	yes	0.055	yes
Systolic blood pressure	0.012	no	0.069	yes	0.055	yes
Temperature	0.022	no	0.077	yes	0.063	yes
Weight	0.235	no	0.223	yes	0.248	yes
pH	0.104	yes	0.102	yes	0.170	yes

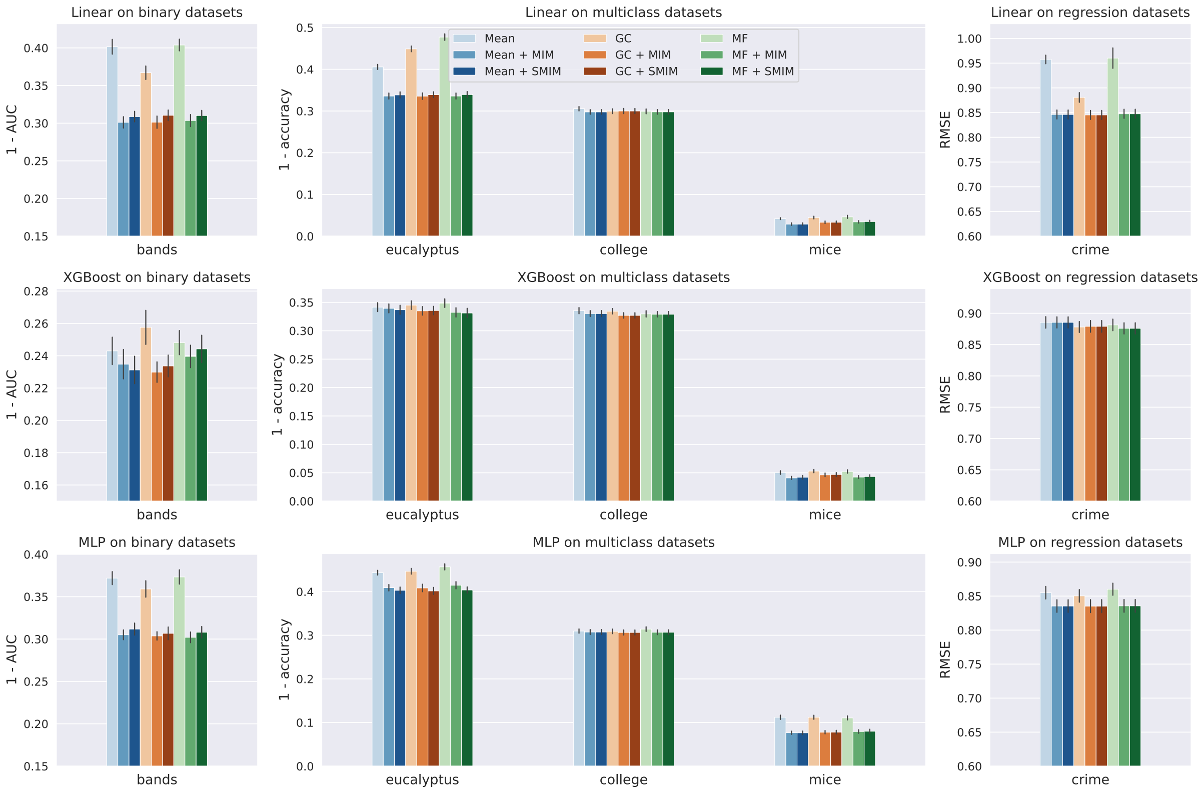

Figure 7: MIM and SMIM performance on OpenML data sets that already have missing values. Both MIM and SMIM improve linear and MLP performance on 4 out of 6 data sets, showing that missing values in real-world data often has informative missingness.

Table 2: OpenML data sets used. Note that for volkert and christine, we remove several features that are either all 0 or are all 0 except a few rows, resulting in the dimensions listed.

OpenML ID	Name	n	p	Task	n_classes
23512	higgs	98050	28	classification	2
1458	arcene	200	10001	classification	2
41150	miniboone	130064	50	classification	2
41145	philippine	5832	309	classification	2
41142	christine	5418	1599	classification	2
1489	phoneme	5404	5	classification	2
6332	bands	540	16	classification	2
41166	volkert	58310	147	classification	10
40498	wine	4898	11	classification	7
41163	dilbert	10000	2000	classification	5
188	eucalyptus	736	9	classification	5
488	college	1161	6	classification	3
537	housing	20640	8	regression	NA
42705	yolanda	400000	100	regression	NA
507	space	3107	6	regression	NA
315	crime	1994	25	regression	NA

C.2 OpenML Real Missing Experiments

We show in Figure 6 that MIM improves performance for EHR prediction tasks, where missing values occur naturally in the data. We also experiment on some additional OpenML data set which already have missing values without masking. We follow the same experimental setup as in Section 5, except that the missing values remain the same in each trial since we are not using a synthetic missing value mask. The results are shown in Figure 7. Like the results on the MIMIC tasks, MIM and SMIM improve performance on all data sets except OpenML's 'college' data set for linear and MLP models. This provides further evidence that missing values are commonly informative in real-world data sets.

C.3 MIMIC

The MIMIC-III data set [20] is a standard data set for building models on electronic health records (EHRs), and has been used by many papers to evaluate machine learning models [2, 9, 13, 32, 40, 58]. Due to its popularity, many tools have been developed to preprocess the raw MIMIC data into forms suitable for data science [17, 35, 46, 51]. We chose to use the mimic3-benchmark [17] to help preprocess our data, creating data sets for the mortality, length of stay (LOS), and phenotype prediction tasks described in Section 6.

For each of the above tasks, the mimic3-benchmark code gathers data from 17 clinical variables as features for the prediction. These 17 features are the same for each task. The feature names are provided in Table 1. For each of the above tasks, the mimic3-benchmark code provides scripts for generating multivariate time series data, with 1 time series per feature per visit that covers a patient's data across their hospital stay. The benchmark code also provides an additional script to generate tabular data using feature engineering on the multivariate time series data to support their logistic regression baselines. Since we study tabular data in this paper, we use this additional preprocessing, and generate 1 feature for each clinical feature in Table 1 corresponding to the mean value across the observed components of the time series. When a time series has no observed time steps, we leave the tabular feature as missing. For each of the resulting features, we compute the missing rate and whether or not the feature has informative missingness based on SMIM, and display the results in Table 1.