scientific reports

Check for updates

OPEN

Flexible imputation toolkit for electronic health records

Alireza Vafaei Sadr[1,2], Jiang Li[3], Wenke Hwang[1], Mohammed Yeasin[4], Ming Wang[5], Harold Lehmann[6,7], Ramin Zand[8,9] & Vida Abedi[1]✉

Missing data in electronic health records (EHRs) poses a significant challenge for analysis. This study introduces Pympute, a comprehensive Python package designed for efficient and robust missing value imputation for EHRs. Pympute's core algorithm, Flexible, intelligently selects the optimal imputation method for each variable based on its characteristics. Pympute offers a comprehensive suite of functionalities. It benchmarks the performance of ten existing machine learning imputation algorithms against Flexible on real-world EHR datasets containing laboratory measurements. Additionally, Pympute facilitates data simulation, generating realistic datasets mimicking real-world data distributions for controlled evaluation of imputation performance. Finally, Pympute investigates how missingness and skewness, influence the selection of optimal imputation algorithms within the Flexible framework. Our findings validate that Pympute's Flexible method significantly improves imputation performance compared to the single model approach. Notably, simulating data solely based on covariance does not accurately reflect real-world selection behavior. Furthermore, skewness in the data distribution prompts Flexible to favor nonlinear imputation models. This study highlights the importance of considering data distribution patterns when selecting imputation algorithms. Pympute addresses this challenge by offering a versatile and user-friendly solution for diverse EHR data scenarios.

Keywords Electronic health records, Missing data, Data imputation, Machine learning

The integration of artificial intelligence (AI) and electronic health records (EHRs) offers transformative opportunities in healthcare, improving diagnostic accuracy, enabling predictive analytics for preventive care, personalizing treatment plans, and optimizing operations such as patient engagement and clinical workflows[1,2] However, the application of EHRs for machine learning-based research is often complicated by missing data, particularly in laboratory variables[3-5] Missing data in EHRs can arise due to various reasons,[6] such as record-keeping errors, workflow, and process issues, variations in data collection protocols, complexities of aggregating information from diverse medical sources, patients seeking treatment at multiple healthcare centers, and inadequate access to healthcare services. Imputing missing data is a common preprocessing strategy for addressing missing data in EHRs. However, the missingness mechanism and pattern (degree of randomness), the percentage of missingness, and various data types and transformations can affect the results of the imputation and ultimately the performance of machine learning models. The choice of the algorithm can also play an important role in imputation performance. Building models on completely observed datasets can result in biased estimates and reduced statistical power, as patterns in EHRs are often missing not-at-random (MNAR). Therefore, imputation methods can play a crucial role in statistical power and model performance.

Several imputation models are available for missing data imputation in electronic health data[6-14] The models' complexity can vary from a simple mean substitution or tree-based methods[15] to expectation maximization[16], complete information maximum likelihood[17], and multiple imputations[18,19] Researchers have also applied highly

[1]Department of Public Health Sciences, College of Medicine, Pennsylvania State University, Hershey, PA, USA. [2]Département de Physique Théorique and Center for Astroparticle Physics, University of Geneva, Geneva, Switzerland. [3]Department of Molecular and Functional Genomics, Weis Center for Research, Geisinger Health System, Danville, PA 17822, USA. [4]Department of EECE, University of Memphis, Memphis, TN 38152, USA. [5]Department of Population and Quantitative Health Sciences, School of Medicine Case Western Reserve University Cleveland, Cleveland, OH, USA. [6]Division of Health Sciences Informatics, Department of Medicine, Johns Hopkins University School of Medicine, Baltimore, MD, USA. [7]Biomedical Informatics and Data Science, Division of General Internal Medicine, Department of Medicine, Johns Hopkins University School of Medicine, Baltimore, MD, USA. [8]Department of Neurology, Milton S. Hershey Medical Center, Penn State Health, Hershey, PA, USA. [9]Neurology Department, Neuroscience Institute, Geisinger Health System, Danville, PA, USA. ✉email: vabedi@pennstatehealth.psu.edu

sophisticated algorithms such as natural language processing[20-22] and graph-based methods[23] for imputation purposes in electronic health data. Additionally, iterative imputation methods, which include imputing missing values and updating the imputation algorithm in each iteration, have also been used in multiple studies[6,24,25] The overall goal of imputation is to help improve the data quality, sample size, and patient representation to further improve model performance or risk prediction, and reduce selection bias caused by complete case analysis.

This study introduces Pympute, a user-friendly, open-source Python package with a Graphical User Interface (GUI) (Supplementary Fig. 1), designed to streamline the imputation process and enhance reproducibility. *Pympute* aligns with core AI readiness concepts and allows for comparing holdout results of ten machine learning-based imputation algorithms and selects the best set of algorithms optimized for each laboratory variable. Unlike conventional imputation methods (e.g., MICE[26] or missForest[27] that apply a single model to all variables, Pympute's Flexible algorithm, guided by the "no-free-lunch" insight[28]—automatically selects the best-performing model for each variable. This per-variable optimization is, to our knowledge, a unique feature that allows each laboratory measurement to be imputed with the algorithm best suited to its distribution and missingness pattern.

We performed a multi-center analysis to evaluate eleven machine learning-based imputation algorithms (ten standard algorithms and as well as the *Pympute* algorithm) across four datasets: MIMIC, Geisinger, Penn State Health, and a simulated dataset derived from Geisinger's EHR data. The datasets vary in terms of laboratory variables, missing data rates, and sample sizes. By employing ten machine learning algorithms, our analysis aims to comprehensively assess their performance and explore how the distribution of real-world data influences optimal imputation algorithm selection. *Pympute* offers cloud-based and GPU-enabled deployment and different error metrics for evaluating imputation performance. This study contributes to the field of EHR-based machine learning research for clinical applications.

Results

We evaluate the performance of *Pympute*, a tool for imputing missing multivariate data, using three different real-world datasets and one simulated dataset. The evaluation includes 10 existing machine learning algorithms, Linear Regression (LR), Ridge Regression (Ridge), Lasso, Elastic Net (ElasticNet), Bayesian Ridge Regression (BRidge), k-Nearest Neighbors (KNN), Multi-Layer Perceptron (MLP)[29], Random Forest (RF)[30], Support Vector Machine (SVM)[31], and XGBoost (XGB)[32], as well as the validation of the Flexible algorithm (Fig. 1a), utilizing two error metrics Root Mean Square Error (RMSE) and Mean Absolute Percentage Error (MAPE). To obtain estimates of these metrics, we employ a holdout approach resembling the missingness. A portion of the data is withheld from the model training process and used exclusively for calculating $MAPE_h$ and $RMSE_h$, where the Subscript *h* denotes holdout-based evaluation. *Pympute* is compatible with both Python and R and can run in parallel on CPU and GPU. The code is open source and publicly available on our laboratory GitHub repository (https://github.com/TheDecodeLab/python-imputation).

Pympute aligns with making AI easier to use (AI readiness). It works in both Python and R and can run faster on special computer parts (CPUs and GPUs) as well as on Google Colab, a cloud computer for large datasets; Pympute can also be used for complex tasks through a web application. Researchers can choose from existing imputation methods or let Pympute choose the best method. Detailed comparative analysis of execution times between GPU and CPU implementations across various dataset sizes and configurations is available in the supplementary materials.

Validation of Pympute on the MIMIC dataset

We first investigated the effectiveness of the algorithms on the MIMIC dataset[33], version 1.4, which consisted of 8,827 selected records with 45 laboratory values (supplementary Table 1). We compared the performances of different imputation algorithms using MAPE and RSME error metrics. Figure 1b; Table 1 present the metric values for each algorithm when applied to the MIMIC dataset.

The results confirm that the Flexible algorithm (*Pympute*) achieved the lowest $MAPE_h$ and the lowest $RMSE_h$. To further compare the performances of the best and second-best (Bridge) algorithms, we calculated the P-values for the differences in $MAPE_h$ and $RMSE_h$ between the Flexible and BRidge algorithms. The P-values were small ($P<0.05$ for $MAPE_h$ and $RMSE_h$), suggesting a significant difference between the two algorithms.

Penn state health data: flexible algorithm consistently selects nonlinear algorithms

We extended our evaluation to the Penn State Health stroke dataset[34], comprising 10,811 ischemic stroke patients with 43 common laboratory values when less than 75% of the data where missing[6]. Table 2 presents the performance metrics for each algorithm when applied to the Penn State Health stroke dataset, with mean $MAPE_h$ and $RMSE_h$ values along with their standard deviations (Fig. 2a).

Our analysis revealed that the Penn State Health variables are skewed (Fig. 2b) and the Flexible model consistently favored nonlinear models over linear models. Specifically, the RF model was the most frequently chosen for 22 laboratory variables, followed by the XGB, MLP, and SVM models, which were selected 8, 3, and 1 times, respectively. Linear models such as the LR and Ridge were chosen 6 and 3 times, respectively.

Figure 2c shows that the Flexible imputation algorithm exhibited superior performance, surpassing the second-best model, RF, based on $MAPE_h$ ($P<0.05$). However, considering $RMSE_h$, the difference between Flexible and RF methods was not statistically significant ($P=0.28$).

We also observed that nonlinear models were preferably chosen by the Flexible model, therefore RF was the most commonly chosen model for 22 laboratory variables and XGB, MLP, and SVM were chosen 8, 3, and 1 times, respectively. The linear models LR and Ridge were selected 6 and 3 times respectively.

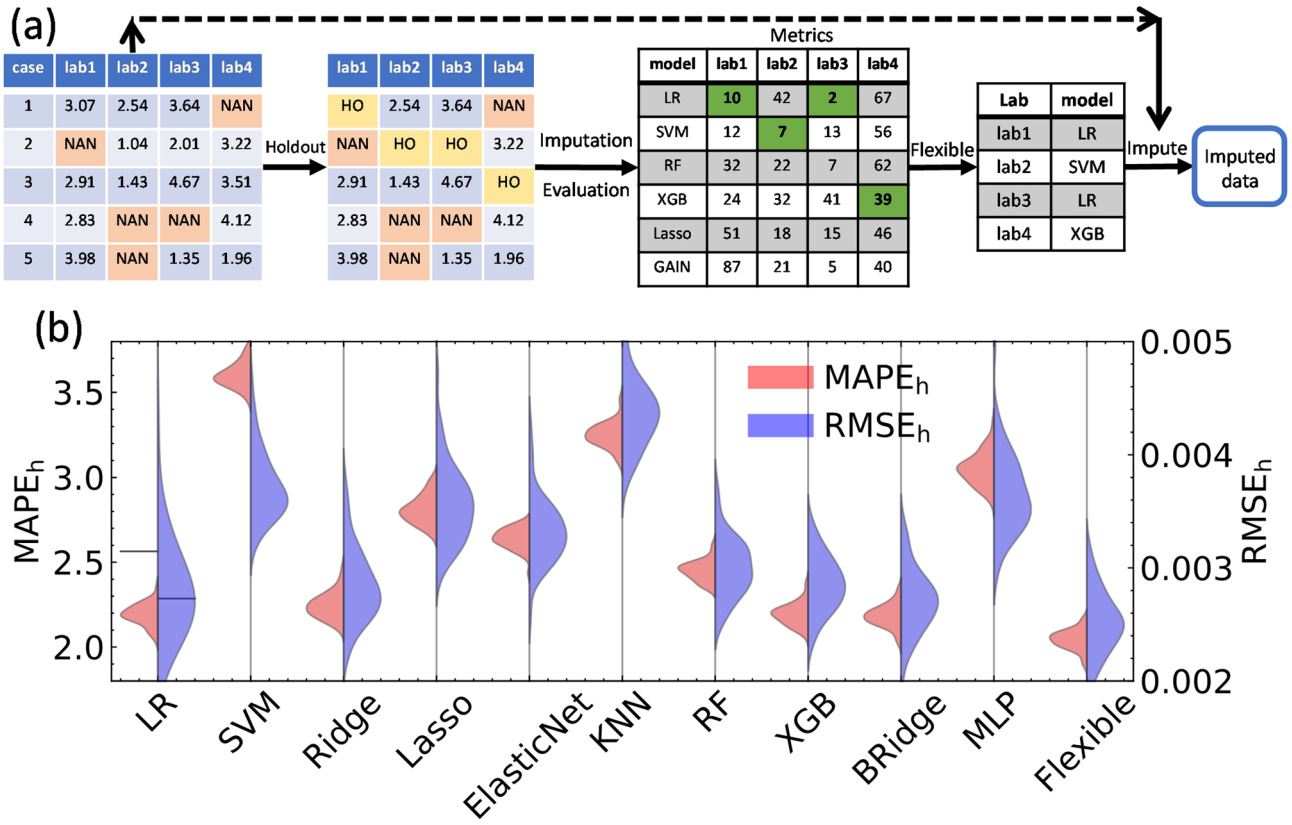

Fig. 1. Flexible Imputation pipeline and preliminary evaluation. (**a**) The Flexible workflow, in which the Pympute algorithm uses holdouts to evaluate all machine-learning algorithms and select the best algorithm for each variable. The default metric for evaluation is the mean absolute percentage error (MAPE). Once the optimal algorithms have been identified, Pympute applies these algorithms to the corresponding variables to generate Flexible results. (**b**) $MAPE_h$ (left axis) and root mean square error ($RMSE_h$, right axis) of the holdout analysis using various machine-learning algorithms on a subset of MIMIC data.

Imputation algorithm	Mean $MAPE_h$ % (SD)	Mean $RMSE_h$ (SD)
BRidge	**2.18 (0.07)**	**0.0027 (0.00028)**
ElasticNet	2.64 (0.06)	0.0033 (0.00029)
KNN	3.25 (0.07)	0.0044 (0.00034)
LR	2.19 (0.06)	0.0029 (0.00073)
Lasso	2.81 (0.08)	0.0036 (0.00034)
MLP	3.05 (0.09)	0.0037 (0.00036)
RF	2.45 (0.06)	0.0030 (0.00026)
Ridge	2.24 (0.08)	0.0028 (0.00032)
SVM	3.60 (0.07)	0.0037 (0.00030)
XGB	2.19 (0.06)	0.0029 (0.00025)
Flexible	**2.04 (0.06)**	**0.0025 (0.00027)**

Table 1. Performance evaluation of imputation algorithms on the MIMIC dataset. This table presents the performance evaluation of various imputation algorithms on the MIMIC dataset. The algorithms were assessed based on two key metrics: mean absolute percentage error ($MAPE_h$) and root mean square error ($RMSE_h$), where the subscript indicates the holdout. The table displays the mean values, along with their corresponding standard deviations (SD) in parentheses. The first and second best-performing algorithms for each metric are highlighted.

Imputation algorithm	Mean $MAPE_h$ % (SD)	Mean $RMSE_h$ (SD)
BRidge	17.67 (7.44)	0.0024 (0.00091)
ElasticNet	35.88 (2.02)	0.0064 (0.00041)
KNN	32.58 (2.97)	0.0039 (0.00037)
LR	15.87 (14.76)	0.0024 (0.00036)
Lasso	36.24 (2.04)	0.0064 (0.00038)
MLP	29.84 (38.76)	0.0022 (0.00027)
RF	**13.42 (1.01)**	**0.0021 (0.00033)**
Ridge	19.44 (5.94)	0.0023 (0.00029)
SVM	21.53 (7.60)	0.0035 (0.00033)
XGB	15.19 (4.05)	0.0022 (0.00025)
Flexible	**12.01 (1.40)**	**0.0020 (0.00026)**

Table 2. Performance evaluation of the imputation algorithms on the Penn state health stroke dataset. This table summarizes the performance of various imputation algorithms on the Penn state health stroke dataset, assessing their effectiveness in handling missing data for 43 common laboratory variables related to ischemic stroke patients. Mean values and standard deviations (SD) are provided for two key metrics, $MAPE_h$ and $RMSE_h$ for holdout analysis.

Geisinger data: the flexible imputation algorithm outperforms the single model approach

We analyzed the results on the Geisinger real-world stroke dataset. Our holdout analysis demonstrated that the Flexible algorithm outperforms all the other algorithms on the Geisinger data. Figure 3a shows a comparison of the metric values for different algorithms, with the Flexible algorithm exhibiting the lowest error rate.

The Ridge algorithm also demonstrated notable performance as the second-best algorithm for the Geisinger data. Its competitive performance suggests that it can be considered an alternative to the Flexible algorithm when applicable. Table 3 presents the $MAPE_h$ and $RMSE_h$ values for each imputation algorithm.

A comparison between the best-performing algorithm, Flexible, and the second-best algorithm, Ridge, was carried out using P-values. The P-value for $MAPE_h$ was found to be $P = 0.014$, indicating a statistically significant difference between the performances of the algorithms. Similarly, a significant improvement in $RMSE_h$ was observed ($P < 0.05$).

Simulating laboratory data based on geisinger EHR data

To replicate a real-world dataset with missing values we utilized real-world Geisinger stroke data from a cohort of 9037 ischemic stroke patients from September 2003 to May 2019[6]. Geisinger is an integrated health system in Pennsylvania, with a catchment area of 3 million. This stroke dataset consists of 45 most commonly ordered laboratory variables, each exhibiting missingness levels below 75% (Fig. 4a; Supplementary Table 2) based on reference study[6].

We employed a multivariate normal distribution (Fig. 4c) for the simulation to follow the observed bivariate normal distribution in the original data. This process resulted in a simulated dataset with dimensions of 9037×45, preserving a similar statistical pattern of missing values as the original dataset (as depicted in Fig. 4a). We recognize that the simulation preserved the normal distribution properties, while acknowledging potential deviations from normality in the observed variables of the original data (Fig. 4b). Our primary emphasis is on assessing the likeness of the covariance matrix between real-world data and simulated data, as depicted in Fig. 4d, demonstrating a notable similarity between the two datasets. We utilized the Mantel test to assess the statistically significant correlation in the correlation matrices of the two datasets, resulting in a coefficient of 0.9974.

The flexible imputation algorithm outperforms other algorithms on simulated data

We assessed the performances of various imputation algorithms on the simulated dataset. Among the evaluated algorithms, the Flexible approach demonstrated the best performance (Fig. 3a and b). Table 4 displays the metric values corresponding to each imputation algorithm utilized for handling missing values and holdouts.

The Flexible algorithm achieved the lowest MAPE, $MAPE_h$, and $RMSE_h$ values, indicating its superior performance compared to that of the other algorithms. The ElasticNet and Lasso algorithms also exhibited competitive results with relatively low RMSE values. In contrast, the SVM algorithm demonstrated the highest metric values, suggesting its limited effectiveness in imputing missing data in the simulated dataset.

To further compare the performances of the best and second-best algorithms, we calculated P-values for the difference in MAPE between the Flexible and ElasticNet algorithms. A significant difference between the two algorithms was observed ($P < 0.05$).

We observed that the average RMSE value for the ElasticNet algorithm was slightly better than that of Lasso, with a difference of 1.4×10^{-5}. Although the RMSE values were quite close, suggesting similar imputation performance between the two algorithms, we further examined the statistical significance by calculating the P-value for comparing their average RMSE values. The resulting P-value of 0.019 indicated statistical significance, implying that the difference between the two algorithms' imputation performances was not due to chance and,

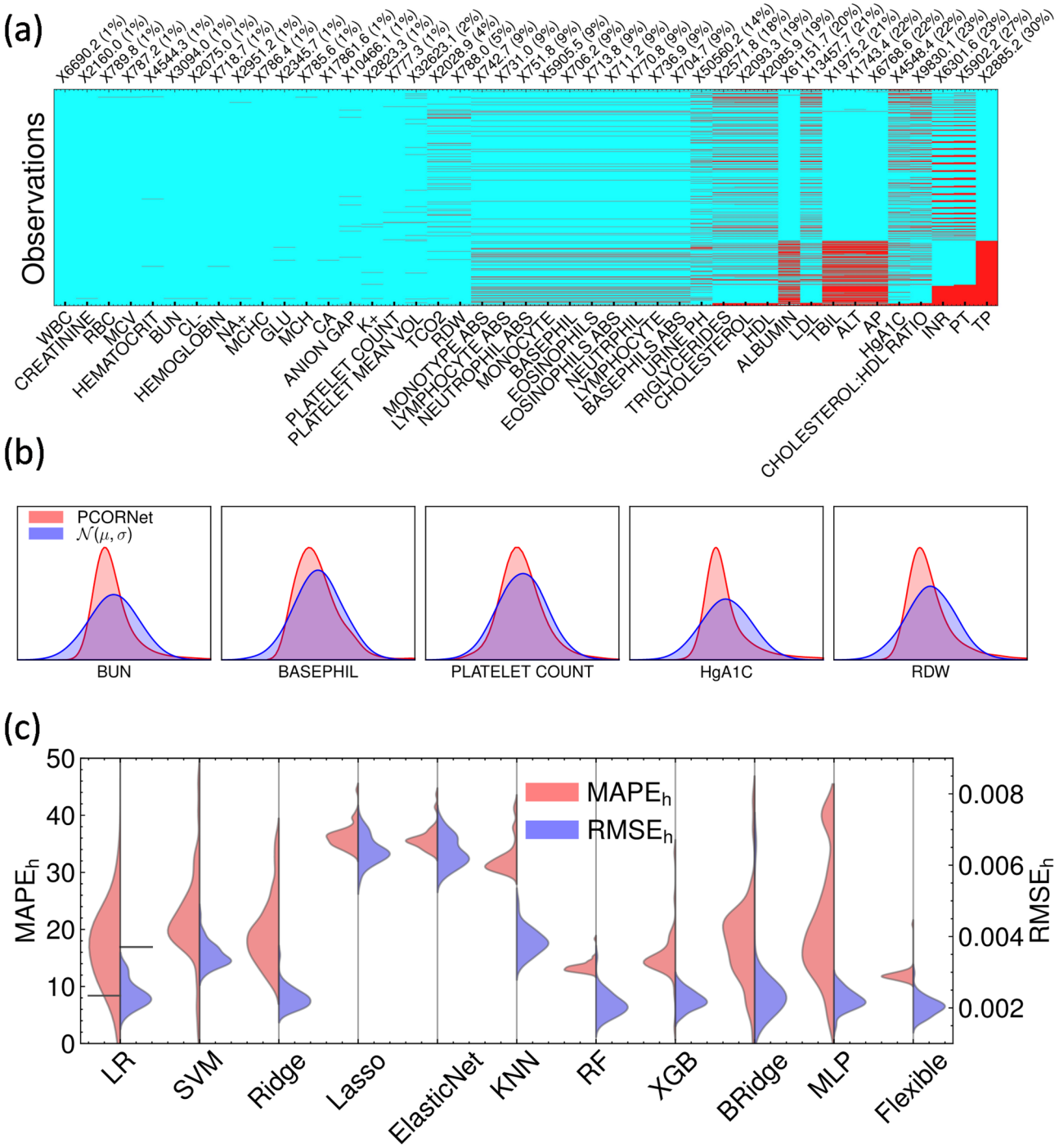

Fig. 2. The Penn State Health data and simulation results. (**a**) The missing pattern in the Penn State Health stroke data, encompassing 10,811 ischemic stroke samples from 43 common laboratories. Red shading denotes missing values. (**b**) presents the distribution of select skewed variables in the Penn State Health data (depicted in red) compared to the normal distribution with the same mean and standard deviation of the corresponding variable. (**c**) Depicts the Mean Absolute Percentage Error (MAPE$_h$, left axis) and root mean square error (RMSE$_h$, right axis) resulting from the holdout analysis utilizing various machine-learning algorithms on Penn State Health data.

indeed, the ElasticNet algorithm showed a superior performance over the Lasso algorithm in terms of RMSE. Additionally, we compared the Flexible and Ridge algorithms, revealing low P-values for both MAPE$_h$ ($P<0.05$) and RMSE$_h$ ($P=0.017$), further supporting the superior performance of the Flexible algorithm.

Notably, we found that a synthetic dataset generated to match only the covariance structure of real data did not replicate the Flexible algorithm's real-world model selections. In other words, an imputation model that

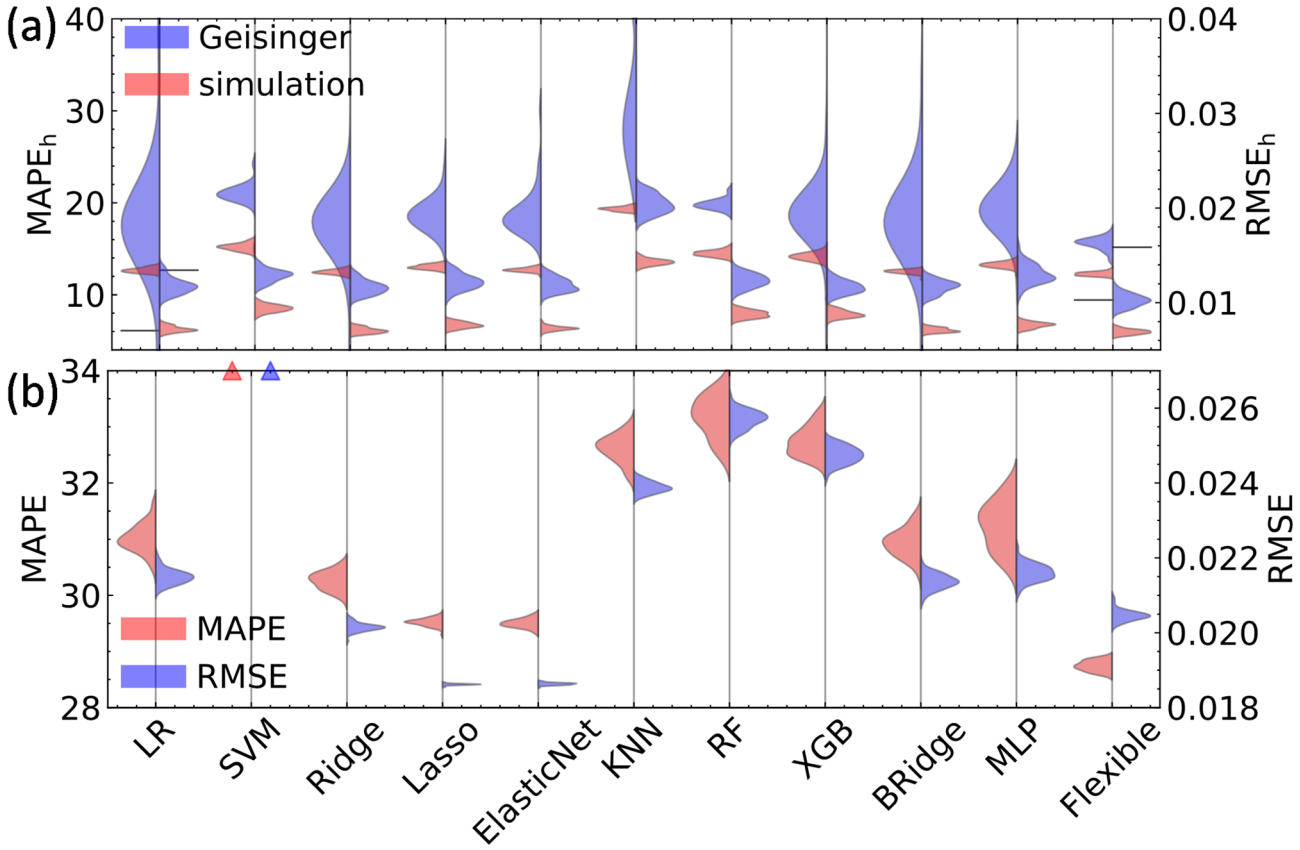

Fig. 3. Comparison of Pympute's imputation performance on the Geisinger stroke data and simulation. (**a**) **Holdout analysis** results for the Geisinger data (blue) and simulation (red) for the mean absolute percentage error ($MAPE_h$, left) and root mean squared error ($RMSE_h$, right) metrics. The Flexible algorithm outperforms all other algorithms in both cases. (**b**) Comparison of the imputed values with the **ground truth** in the simulation data. The figure presents a visual assessment of imputation performance using two metrics: Mean Absolute Percentage Error (MAPE) in red and Root Mean Square Error (RMSE) in blue. The Flexible algorithm demonstrates favorable imputation accuracy on the simulated data. However, for RMSE, Lasso and ElasticNet outperform the Flexible algorithm due to the optimization criteria being based on $MAPE_h$.

appeared optimal in simulated data was often suboptimal on actual real-world EHR data. This suggests that commonly used simulation approaches may overlook critical distributional nuances (e.g., skewness or MAR/MNAR patterns), a limitation that our results bring to light.**The Influence of the Missing Level and Distribution Skewness on the Imputation Performance and Algorithm Selection on the Geisinger Stroke Data**.

We examined the influence of the missing data level and distribution skewness on imputation error and the algorithm selection process in the Flexible method. Figure 5a demonstrates the poor correlation between the missing level and $MAPE_h$ for both the simulated data ($r=0.08$) and the Geisinger stroke data ($r=0.24$). The P-value of 0.94 supports a nonsignificant difference between Geisinger and simulation data. While this correlation is small, it is expected that more missing data leads to higher errors. Figure 5b shows a weak (more than 0.3 but less than 0.5) correlation between skewness and $MAPE_h$ in the Geisinger data ($r=0.39$). This suggests that skewness can affect the results, as more skewed distributions are more difficult to impute. This outcome is different from the simulation data ($P<0.05$), which is derived from a normal distribution and therefore has a skewness that is close to zero, resulting in a non-significant correlation.

We also investigated how the algorithm selection process may choose different algorithms for the simulated data versus the real data. Figure 5c displays the selection count of algorithms selected by the Flexible method for both the Geisinger and simulated data. Our results show that the simulated data is restricted to linear algorithms and does not prefer any of the nonlinear algorithms. The Flexible imputation algorithm often selected nonlinear algorithms such as random forests and XGB when tested on the Geisinger data. Our findings indicated that regression-based methods, such as ElasticNet, LR, Lasso, and Ridge, consistently demonstrated favorable performance in the holdout analysis of the simulated data, particularly when the data distribution showed no or minimal skewness.

We selected four laboratory variables (HgA1C, ALT, LDL, and cholesterol) with varying degrees of skewness. The distributions of these variables, along with their respective skewness values, are presented in Fig. 6. Larger errors indicated the increased uncertainty in the predicted variables due to the skewness of the distribution. To provide a more rigorous assessment of imputation performance, we conducted t-tests on the $MAPE_h$ values

Imputation algorithm	Mean $MAPE_h$ % (SD)	Mean $RMSE_h$ (SD)
BRidge	18.84 (7.98)	0.0117 (0.00070)
ElasticNet	18.79 (2.45)	0.0119 (0.00074)
KNN	44.24 (9.82)	0.0204 (0.00099)
LR	18.08 (8.35)	0.0118 (0.00084)
Lasso	18.32 (3.00)	0.0122 (0.00082)
MLP	18.06 (4.99)	0.0130 (0.00103)
RF	19.85 (0.59)	0.0123 (0.00074)
Ridge	**17.72 (6.00)**	**0.0116 (0.00073)**
SVM	20.94 (1.21)	0.0129 (0.00067)
XGB	19.18 (5.32)	0.0117 (0.00068)
Flexible	**15.55 (0.70)**	**0.0102 (0.00066)**

Table 3. Performance evaluation of imputation algorithms on geisinger stroke data. This table presents the performance evaluation of different imputation algorithms on the geisinger stroke dataset. The evaluation is based on two key metrics: mean absolute percentage error (MAPE) and root mean square error (RMSE) with the subscript indicating the holdout. The table displays the mean values for each algorithm, along with their corresponding standard deviations (SD) in parentheses. The first and second best-performing algorithms for each metric are highlighted.

for all pairs of machine learning algorithms. The results are depicted in Fig. 6, demonstrating significant differences in $MAPE_h$ values between various algorithm combinations. The comprehensive results for each imputation algorithm concerning individual laboratory variables are presented in Supplementary Tables 4 and Supplementary Fig. 2.

Discussion

In this study, we evaluated the performance of ten machine learning-based imputation algorithms to address missing data in electronic health record datasets.

Validation of the flexible algorithm

Our results demonstrate that the Flexible imputation algorithm outperforms other algorithms in terms of MAPE on MIMIC, simulated data, and the two additional real-world EHR-based datasets (Geisinger and Penn State Health). The results from the holdout and ground truth analysis on the studied datasets, which utilized the missing pattern in the Geisinger dataset, indicate that employing the holdout analysis approach can lead to minimal errors.

Factors influencing algorithm selection and their performance

Various factors, including missingness pattern (monotonic or not), mechanism -Missing Completely at Random (MCAR), Missing at Random (MAR), Missing Not at Random (MNAR)-, missingness level, sample size, and data transformation, exert notable influences on imputation performance and algorithm choice. Greater missingness, smaller sample sizes, and MNAR scenarios amplify prediction uncertainty due to limited observed data. Data adhering to a multivariate normal distribution are predicted with less uncertainty. However, real-world data often exhibit MNAR patterns, as demonstrated in this study through the application of MNAR masks to simulated data.

The distinct imputation methods react differently to non-normal data distributions. While multiple imputation might counterbalance larger imputation errors from nonnormality, comparative assessments across regression-based (e.g., linear regression, Lasso, Ridge) and nonregression-based (e.g., KNN[35], Random Forest[27] algorithms remain underexplored. We observed that data skewness strongly influences algorithm preference: in highly skewed lab variables, Flexible almost exclusively selected nonlinear models (Random Forest, XGBoost, etc.), whereas for nearly variables with a normal distribution, the model often chose linear regression-based methods. This aligns with concerns in the literature that imputing skewed data with normal-model assumptions can introduce bias[36] Our analysis is the first to empirically demonstrate that in the presence of skewed distributions, nonlinear imputation models yield better accuracy, validating the need to tailor the method to the data's underlying distribution.

Prioritizing classic machine learning algorithms

The application of deep learning algorithms[37-42], such as GAIN[38], for tabular data imputation has been met with challenges, as evidenced by benchmarking studies[43-46]. These studies consistently show that classic machine learning algorithms outperform deep-learning algorithms on diverse real-world tabular datasets. Given these findings and the "no-free-lunch" theorem, which emphasizes the absence of a universally optimal algorithm, we chose to focus on classic machine learning algorithms, like ElasticNet, LR, Lasso, and Ridge, for tabular data imputation in our study. While deep learning holds promise in other domains, we believe the current state of research warrants prioritizing well-established and efficient algorithms for our specific tabular datasets,

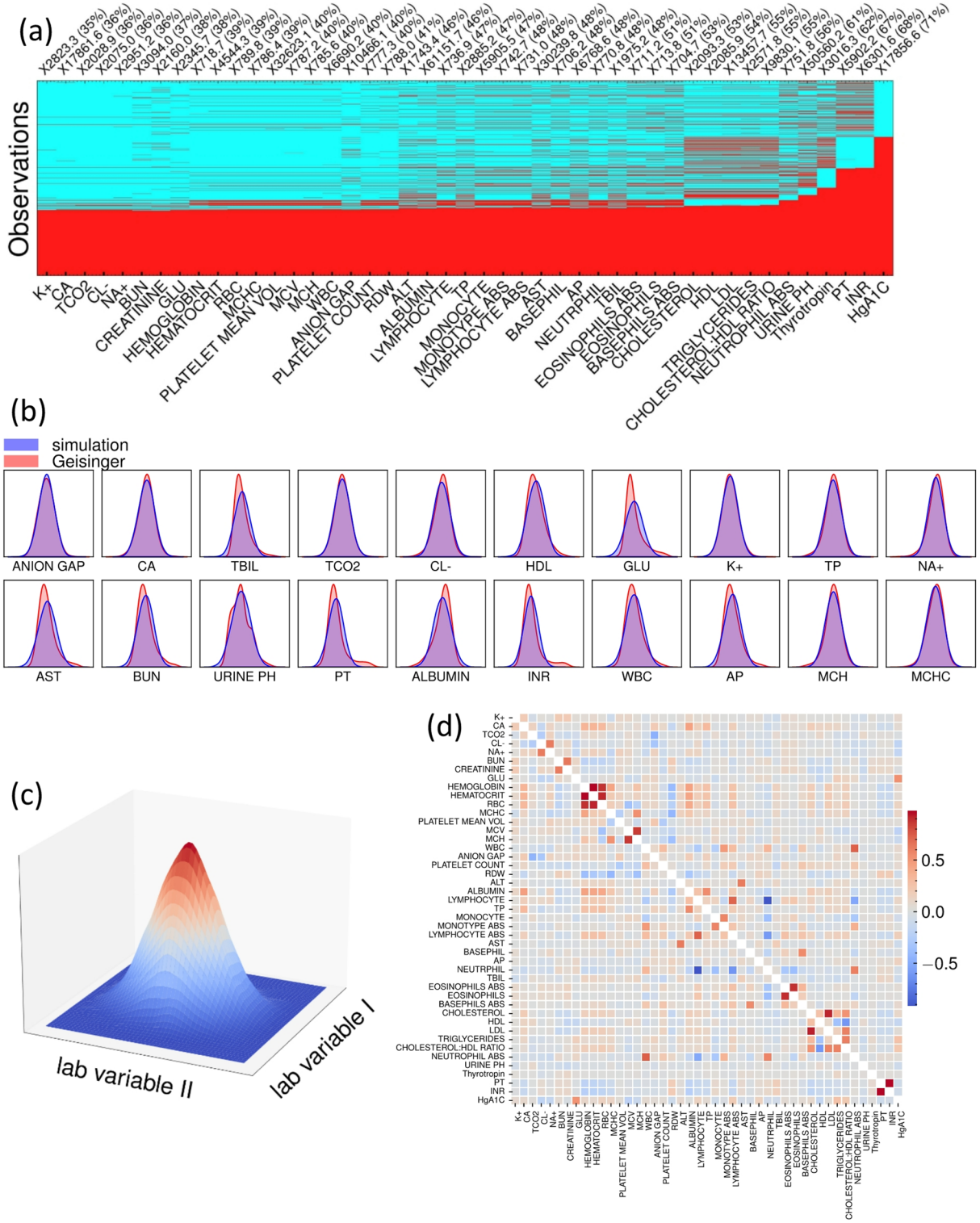

Fig. 4. The Geisinger stroke data and simulation results. (**a**) displays the missing pattern in the Geisinger data, which includes 9037 ischemic stroke samples from 45 common laboratories. Red indicates the missing values. (**b**) compares the distributions of real (blue) and simulated (red) data for a subset of laboratory variables. The simulation is based on a normal distribution assumption, leading to larger differences when a variable (e.g., GLU) deviates significantly from Normal distribution. (**c**) displays a schematic view of the probability density function of the multivariate normal distribution based on point estimators derived from the observed data. This figure is made using Python 3.11.5 and Matplotlib 3.7.2 employing the "plot_surface" function. (**d**) illustrates the covariance based on the correlation coefficient matrix and variance of each laboratory variable. Using the 'mvrnorm' function from the R 'MASS' package, we generated a dataset with dimensions of 9037×45, representing 9037 samples and 45 laboratory variables, based on the mean and covariance values.

Imputation algorithm	Mean MAPE % (SD)	Mean RMSE (SD)	Mean $MAPE_h$ % (SD)	Mean $RMSE_h$ (SD)
BRidge	30.93 (0.25)	0.021 (0.00021)	12.51 (0.21)	0.0071 (0.00024)
ElasticNet	**29.50 (0.08)**	**0.019 (0.00030)**	12.71 (0.22)	0.0073 (0.00025)
KNN	32.61 (0.24)	0.024 (0.00015)	19.33 (0.22)	0.0143 (0.00033)
LR	31.01 (0.25)	0.022 (0.00020)	12.66 (0.25)	0.0072 (0.00032)
Lasso	29.52 (0.07)	**0.019 (0.00023)**	13.03 (0.25)	0.0076 (0.00031)
MLP	31.24 (0.43)	0.022 (0.00023)	13.23 (0.33)	0.0076 (0.00031)
RF	33.18 (0.38)	0.026 (0.00024)	14.49 (0.33)	0.0088 (0.00039)
Ridge	30.26 (0.16)	0.020 (0.00011)	**12.47 (0.25)**	**0.0070 (0.00029)**
SVM	39.09 (0.53)	0.039 (0.00026)	15.18 (0.38)	0.0094 (0.00039)
XGB	32.74 (0.26)	0.025 (0.00023)	14.05 (0.72)	0.0088 (0.00038)
Flexible	**28.74 (0.09)**	0.021 (0.00015)	**12.27 (0.21)**	**0.0069 (0.00027)**

Table 4. Performance evaluation of imputation algorithms on simulated data. This table presents the performance evaluation of different imputation algorithms on the simulated data. The algorithms were assessed based on two key metrics: mean absolute percentage error (MAPE) and root mean square error (RMSE) with the subscript indicating holdout. The table displays the mean values for each algorithm, along with their corresponding standard deviations (SD) in parentheses. The first and second best-performing algorithms for each metric are highlighted.

extracted from structured EHR datasets. However, we acknowledge that further exploration in deep learning and transform-based algorithms for tabular data imputation may lead to specialized architectures in the future.

Data transformation

In this study, we employed a MinMax transformation to standardize the range of data per variable within the [0, 1] window. This approach improved the results by reducing the impact of varying scales and enhancing result stability. Notably, both normalization and scaling methods can alter the original relationships between variables. Min-max scaling and z-score transformation, commonly used scaling methods, restrict data to specific ranges. While min-max scaling suppresses outlier influence by minimizing standard deviation, z-score transformation preserves outlier impact.

Our pre-imputation data scaling aimed to maintain variable skewness and correlation. Applying an inverse transformation might induce variable normality but could also alter the correlation between prediction and outcome variables[36,47]. Such transformations might not always ensure normality, potentially affecting bivariate relationships assumed by imputation methods.

Study strengths and limitations

Our study possesses several strengths contributing to robustness. This study included a multicenter analysis that compared synthetic and real data across two distinct patient cohorts: 8,827 ICU patients, and 9,037 and 10,811 stroke patients from Geisinger and Penn State Health Data respectively.

It's worth noting that alternative strategies like the GAMLSS method[48,49] exhibit adaptability by accommodating diverse distribution parameters, and the ImputeRobust[50] package offers an array of *mice* methods[47] tailored to continuous data scenarios. This study comprehensively compares machine learning-based imputation algorithms on real-world and simulated datasets, emphasizing algorithm choice and accounting for variable skewness. The development of an accelerated user-friendly imputation tool facilitates efficient EHR-based laboratory data imputation. A detailed comparison of Pympute with the widely used MICE package is provided in the supplementary materials (Supplementary Table 1) to facilitate informed selection based on specific research requirements.

Our study has several limitations. The 75% threshold for missing values is based on a previous study[6], and while sensitivity analysis is an important direction, its complexity merits further investigation. Bayesian skepticism towards P values, especially in simulated environments, is noted, and consideration of different setups for skewness, missing percentages, and others remains a potential avenue for future exploration.

Furthermore, the fixed holdout set size of 50, while supported by sensitivity analysis (Supplementary Fig. 4), might not be optimal for all datasets. A more adaptive approach to holdout set size selection could be explored in future research. Additionally, the random holdout approach may not fully capture real-world missing data patterns, potentially affecting the generalizability of our findings.

Finally, in cases where data follow multivariate Gaussian patterns with linear relationships, the FCS based method may introduce bias by overfitting observed patterns. Although we addressed this concern through multiple imputations to capture uncertainty, further investigation may be needed to ensure robust imputation in such contexts.

Methodological innovation

We present Pympute, an open-source, EHR-focused toolkit that, to our knowledge, is the first to adaptively choose the best imputation model for each variable before any downstream analysis. Traditional approaches apply one method to all features; Pympute instead evaluates multiple algorithms on a small hold-out slice of

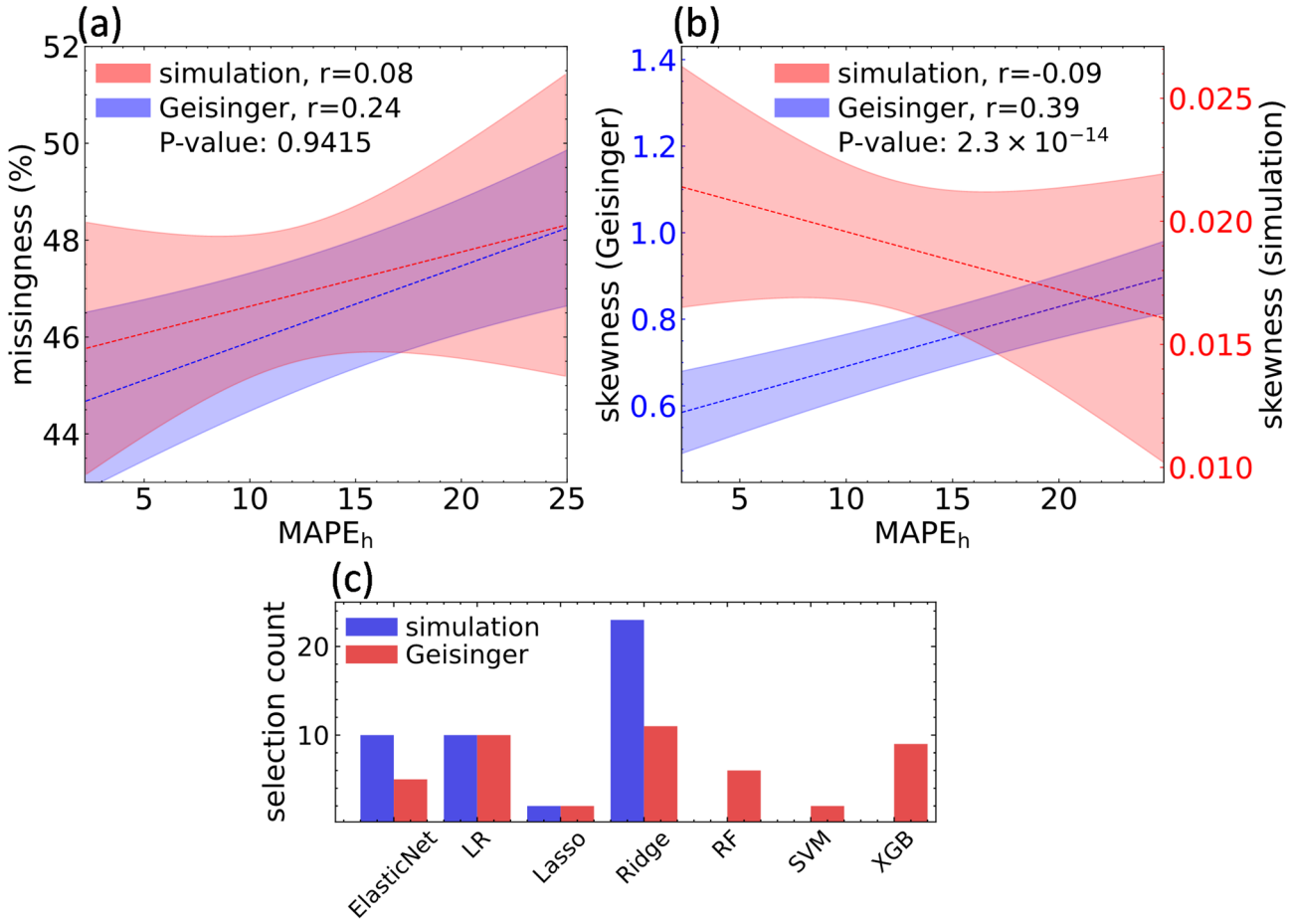

Fig. 5. The influence of missingness and skewness on imputation error. (**a**) illustrates the weak correlation between missingness and MAPE for both simulated data ($r=0.08$) and Geisinger stroke data ($r=0.24$). It is expected that more missing data leads to higher errors. (**b**) depicts a weak positive correlation (0.2 to 0.4) between skewness and MAPE in Geisinger data ($r=0.39$) and a very weak negative correlation (-0.2 to 0.0) for simulated data (r=-0.09). The simulation data, represented by red ticks on the double y-axis, is based on a Normal distribution, resulting in a smaller range of skewness compared to Geisinger (blue ticks). (**c**) displays the frequency of algorithms selected by the Flexible method for both Geisinger data (red) and simulated data (blue). The simulated data is limited to linear algorithms, while the Geisinger data utilizes nonlinear algorithms such as random forests.

observed data and selects the lowest-error model per variable. Tested on three hospital datasets and a simulation study, this strategy consistently improves imputation accuracy and clarifies when simple linear models suffice versus when skewed or MNAR data require nonlinear methods.

This design is motivated by the observation that no single imputation strategy dominates across all scenarios[28] By letting the data themselves dictate the model choice variable-by-variable, Pympute embodies the 'no one-size-fits-all' principle and delivers accuracy gains we demonstrate across three real-world hospital datasets.

Recent work by Williamson & Huang (2024)[51] introduced a Super-Learner–based variable-selection procedure that operates after multiple imputation. By contrast, Pympute addresses the upstream challenge of model selection during imputation. The two methods, therefore, solve complementary stages of the pipeline. The GPU acceleration is included only to expedite large-scale EHR processing; the core contribution is the adaptive, per-variable imputation framework and the empirical guidance it provides for real-world data.

Future directions

In the future, we will extend our Flexible imputation strategy to data generated from clinical trials with distinct characteristics and data distributions. Additionally, we are also exploring deep learning generative algorithms, and incorporating missingness patterns in imputation for more comprehensive simulations.

Conclusions

In conclusion, our study provides a comprehensive evaluation of machine learning-based imputation algorithms for handling missing data in EHR datasets. The Flexible imputation algorithm shows enhanced performance across various datasets, including real-world EHR datasets (MIMIC, Geisinger, and Penn State

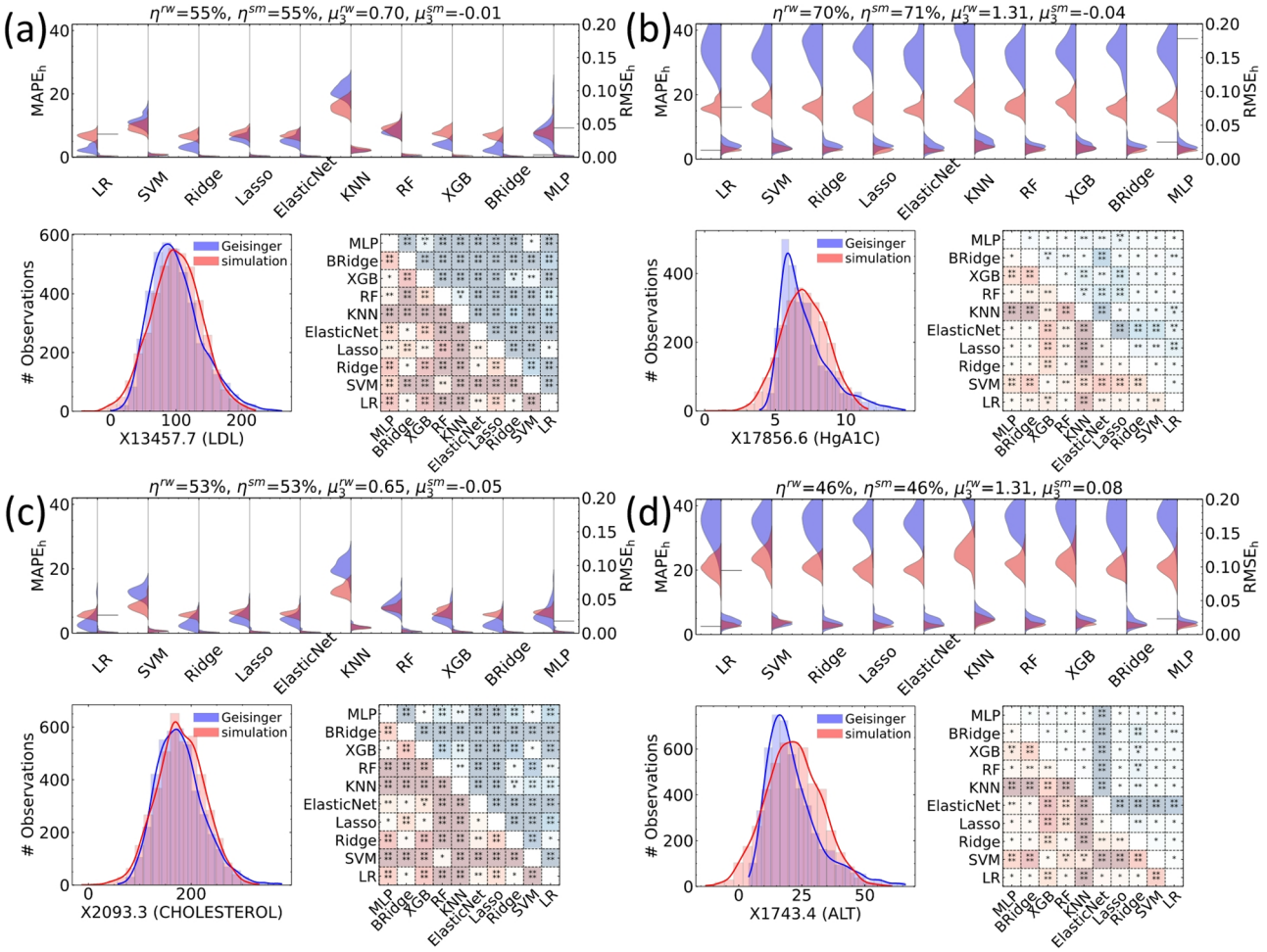

Fig. 6. Detailed comparison of algorithms for four laboratory variables, with varying missingness and skewness. In each panel, the three plots present $MAPE_h$ and $RMSE_h$ comparison, the distribution of real and simulated data, and, the statistical comparison of MAPE values for all machine-learning algorithm pairs using p-values. The stars in the table are corresponding to p-values, indicating the statistical significance of differences between the imputation performances. The missingness (η) and skewness (μ) values are also displayed at the top of each panel. In all plots Geisinger and the corresponding simulation data are indicated by blue and red, respectively. The findings indicate that larger skewness leads to larger errors and a greater disparity in imputation performance between the simulation and Geisinger stroke data.

Health) and simulated data. We elucidate the impact of data distribution and skewness on algorithm selection and performance, revealing that the normality of data significantly influences imputation outcomes and algorithm selection preference. Our study has strengths in its multicenter analysis and comparison of real-world and simulated datasets, and its limitations and future directions underscore the ongoing pursuit of refining imputation strategies for diverse healthcare data scenarios.

Methods

The study protocol and laboratory data from Geisinger and Penn State were reviewed and approved by the Geisinger and Penn State College of Medicine Institutional Review Board (IRB), which determined that the study qualified as exempt from human subjects' research requirements due to the use of fully de-identified data. As such, Informed consent is waived by the Ethics committee at Geisinger IRB and Penn State College of Medicine IRB. All methods were carried out in accordance with relevant guidelines and regulations.

Pympute

Flexible or Flexibly employs an iterative imputation approach based on Fully Conditional Specification (FCS), which handles mixed variable types (binary, categorical, and continuous) through separate conditional models. This method models each variable with missing values as a function of the other observed variables in the dataset. Unlike Joint Modeling (JM), which requires all variables to follow a multivariate Gaussian distribution, FCS accommodates complex dependencies between different variable types. The imputation process begins by replacing missing values with initial estimates, often obtained through imputation methods like mean or median

imputation. To mitigate potential biases and capture probable variability range in the imputed results, Flexible or Flexibly performs per-variable multiple imputations, generating several complete datasets that are analyzed separately and pooled to produce final estimates, thereby accounting for imputation uncertainty and reducing overfitting. This approach maintains data coherence while accommodating complex dependencies between different variable types.

In this study, we used a random sample of the existing values as an initial estimation. Subsequently, an iterative cycle commences. In each iteration, imputation models are constructed for each variable with missing values, using the other variables as predictors. Missing values are then replaced with predictions from these models. This process continues to replace the estimations until convergence, typically determined by a predefined threshold for the maximum absolute change in imputed values between successive iterations. In this study, a convergence threshold of 10^{-4} was employed.

Pympute is a tool that utilizes multiple machine learning algorithms (not as an ensemble model) including Linear Regression (LR), Ridge Regression (Ridge), Lasso, Elastic Net (ElasticNet), Bayesian Ridge Regression (BRidge), k-Nearest Neighbors (KNN), Multi-Layer Perceptron (MLP)[29], Random Forest (RF)[30], Support Vector Machine (SVM)[31], and XGBoost (XGB)[32], as well as the Flexible algorithm, which selects the best algorithm for each variable based on the analysis of holdout values. In our study, we employed two distinct computational frameworks for model development, contingent on the processing unit in use. For models that were computed using CPUs, we utilized the Scikit-learn library[52], a popular open-source tool in Python offering a range of supervised and unsupervised learning algorithms. Conversely, for models that leveraged the power of GPUs, we employed the cuDF library. CuDF is a GPU-accelerated data manipulation library integrated into the RAPIDS data science framework. This bifurcated approach allowed us to optimize computational efficiency across different hardware configurations. A short description of these machine learning algorithms is provided in the supplementary materials. The traditional train-test splitting is not applicable as models iteratively impute/update missing values without direct access to ground truth. Instead, a hold-out analysis is conducted to assess imputation performance for each variable.

Per-variable imputation

To determine the best imputation model for each feature, Pympute's Flexible algorithm uses an internal hold-out evaluation. Specifically, for each variable with missing values, we randomly withhold a small subset of its observed entries (e.g., 50 data points, as justified by a sensitivity analysis) to serve as a pseudo-missing hold-out set. We then fit each candidate imputation model using the remaining observed data for that variable (with all other variables as predictors) and impute the held-out values. By comparing the imputed values to the true values in the hold-out set, we compute an error (using MAPE by default, and RMSE) for each model. The algorithm selects the model with the lowest error on this hold-out as the optimal imputation method for that variable (Fig. 1a). This selected model is subsequently used to impute all missing values for the variable.

To account for uncertainty in the imputation, Pympute can perform multiple imputations by repeating the above procedure multiple times. In our implementation, we generated five complete datasets using different random initializations for the FCS algorithm. Each iteration of the FCS process yields one imputed dataset once convergence is reached. We then pool the results across these five imputed datasets to produce final estimates – for example, by averaging imputed values or applying Rubin's rules[53] to any downstream analysis – thereby incorporating the variance between imputations. This multiple-imputation approach ensures that uncertainties in predicted values are reflected in the final model.

Experimental design

In this study, we compare the performance of *Pympute* using two metrics: Root Mean Square Error (RMSE) and Mean Absolute Percentage Error (MAPE):

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

$$MAPE = \frac{100\%}{n}\sum_{i=1}^{n}\left|\frac{y_i - \hat{y}_i}{\hat{y}_i}\right|$$

RMSE, computed on normalized data, gauges the disparity between predicted and actual values, $\widehat{y}_i$. A lower RMSE signifies a superior fit; however, the dataset range impacts the interpretation. For MAPE, values below 20% are considered indicative of a good fit.

In all experiments, the datasets were scaled to a range of 0 to 1 prior to imputation, and outliers, defined as values beyond 3 times the interquartile range (IQR), were removed. The imputation process was repeated 50 times for each algorithm, with 20 iterations per imputation. A total of 50 holdouts were used in all experiments and were determined based on previous research by Jiang et al.[6]. For each imputation algorithm, we performed a tuning step to identify the optimal configuration. Due to our computational resource limitations, we employed an exhaustive grid search approach to tuning the algorithms. The grid search involved exploring at least six given configurations, encompassing various hyperparameter combinations. For all statistical tests, we conducted a P-value analysis, including an independent t-test with the Levene test to assess variance equality.

Flexible algorithm

The cornerstone of Pympute's functionality is the Flexible imputation algorithm, which leverages a powerful technique called fully conditional specification (FCS) to tackle missing values in electronic health record (EHR)

data. FCS operates on a variable-by-variable basis, essentially addressing missingness for each laboratory variable independently.

Here's how Flexible imputation with FCS works: The algorithm first employs a holdout analysis for each variable with missing values. Pympute allows researchers to choose from various evaluation metrics for model selection. In this study, we utilized $MAPE_h$, a common metric in forecasting tasks, to assess the performance of each imputation algorithm on the unseen holdout set. $MAPE_h$ measures the average absolute percentage error between predicted and actual values.

Finally, based on the $MAPE_h$ scores obtained from the holdout analysis, the Flexible algorithm identifies the most effective machine learning algorithm configuration for each variable. This configuration essentially represents the imputation model that minimizes the error on unseen data, leading to more accurate data imputation. Figure 1a provides a schematic view of the Flexible algorithm's workflow, illustrating the key steps involved in selecting the optimal imputation model for each variable using FCS.

Dataset simulation

We propose a simulation strategy using a probability density function of multivariate normal distribution (MVN) for a random vector $x \in R^d$ in the observed data.

$$N(x|\mu, \Sigma) \triangleq \frac{1}{(2\pi)^{d/2}|\Sigma|^{1/2}} exp\left[-\frac{1}{2}(x-\mu)^T \Sigma^{-1}(x-\mu)\right]$$

The simulated data follows a bivariate normal distribution, with mean vector and variance-covariance matrix Σ. To create the simulated dataset, we utilized the 'cor2cov' function from the R 'Jwileymisc' package to generate a matrix of covariance (Σ) based on the correlation coefficient matrix () and variance () of each laboratory variable and applied the 'mvrnorm' function from the R 'MASS' package.

We then applied the mask of missingness from real-world data to the simulated data. The simulated data were subjected to holdout testing, in which values were randomly dropped in each run, and a missing value test using the missing pattern in the real-world Geisinger dataset.

Data availability

Data and Code Availability: The data availability for this study adheres to the principles of transparency and scientific reproducibility. Detailed information regarding the accessibility of the datasets and code resources is provided below: MIMIC Data: The MIMIC dataset, version 1.4, utilized in this study is publicly available and can be accessed online6. Simulated Data: The simulated dataset, which emulates the missing values observed in the real-world Geisinger data, will be made available upon request, request can be made to Jiang Li (jli@geisinger.edu). The simulated data is accessible via the GitHub repository at https://github.com/TheDecodeLab/simulation-of-laboratory-variables-using-Multivariate-Gaussian-approach.Geisinger Data: Due to privacy and confidentiality concerns, we are unable to publicly share the Geisinger real-world dataset. The data was collected from a large integrated healthcare system encompassing multiple hospitals in the United States, necessitating strict adherence to privacy regulations and ethical considerations. However, data can be shared upon execution of a data-sharing agreement; interested parties can contact Jiang Li (jli@geisinger.edu) to request access.Penn State Health Data: Due to privacy and confidentiality concerns, we are also unable to publicly share the Penn State Health real-world dataset. However, data can be shared upon execution of a data-sharing agreement; interested parties can contact Vida Abedi or Wenke Hwang (vabedi@pennstatehealth.psu.edu or whwang@pennstatehealth.psu.edu) to request access.Scripts and Pympute Package: To promote transparency and enable reproducibility, the code used in this research and the Pympute package is released as open-source resources on GitHub (https://github.com/TheDecodeLab) (upon publication). Interested researchers can access the codebase, utilize Pympute for their imputation tasks, and replicate the methodology employed in this study.

Received: 11 April 2024; Accepted: 12 May 2025
Published online: 17 May 2025

References

1. Shah, P. et al. Artificial intelligence and machine learning in clinical development: a translational perspective. *Npj Digit. Med.* **2**, 1–5 (2019).
2. Ashton, J. J., Young, A., Johnson, M. J. & Beattie, R. M. Using machine learning to impact on long-term clinical care: principles, challenges, and practicalities. *Pediatr. Res.* **93**, 324–333 (2023).
3. Huang, S. C., Pareek, A., Seyyedi, S., Banerjee, I. & Lungren, M. P. Fusion of medical imaging and electronic health records using deep learning: a systematic review and implementation guidelines. *Npj Digit. Med.* **3**, 1–9 (2020).
4. Lin, W. et al. Predicting Alzheimer's disease conversion from mild cognitive impairment using an extreme learning Machine-Based grading method with multimodal data. *Front. Aging Neurosci.* **12**, 77 (2020).
5. Kline, A. et al. Multimodal machine learning in precision health: A scoping review. *Npj Digit. Med.* **5**, 1–14 (2022).
6. Li, J. et al. Imputation of missing values for electronic health record laboratory data. *Npj Digit. Med.* **4**, 1–14 (2021).
7. Khurshid, S. et al. Cohort design and natural Language processing to reduce bias in electronic health records research. *Npj Digit. Med.* **5**, 1–14 (2022).
8. Garriga, R. et al. Machine learning model to predict mental health crises from electronic health records. *Nat. Med.* **28**, 1240–1248 (2022).
9. Zhou, Y. H., Saghapour, E. & ImputEHR A visualization tool of imputation for the prediction of biomedical data. *Frontiers Genetics* **12**, (2021).
10. Amrollahi, F., Shashikumar, S. P., Holder, A. L. & Nemati, S. Leveraging clinical data across healthcare institutions for continual learning of predictive risk models. *Sci. Rep.* **12**, 8380 (2022).
11. Rajkomar, A. et al. Scalable and accurate deep learning with electronic health records. *Npj Digit. Med.* **1**, 1–10 (2018).

12. Acosta, J. N., Falcone, G. J., Rajpurkar, P. & Topol, E. J. Multimodal biomedical AI. *Nat. Med.* **28**, 1773–1784 (2022).
13. Multimodal data integration. Improves immunotherapy response prediction. *Nat. Cancer.* **3**, 1149–1150 (2022).
14. Li, J. et al. Predicting mortality among ischemic stroke patients using pathways-derived polygenic risk scores. *Sci. Rep.* **12**, 12358 (2022).
15. Rahman, G. & Islam, Z. Australian Computer Society, Inc., AUS,. A decision tree-based missing value imputation technique for data pre-processing. in *Proceedings of the Ninth Australasian Data Mining Conference - Volume 121* vol. 121 41–50 (2011).
16. A Comparison of Imputation Techniques for Handling Missing Data -, Musil, C. M., Warner, C. B., Yobas, P. K. & Jones, S. L. (2002). https://journals.sagepub.com/doi/10.1177/019394502762477004
17. Enders, C. K. A primer on maximum likelihood algorithms available for use with missing data. *Struct. Equation Modeling: Multidisciplinary J.* **8**, 128–141 (2001).
18. The use and reporting of multiple imputation. in medical research – a review - Mackinnon – 2010 - Journal of Internal Medicine - Wiley Online Library. https://onlinelibrary.wiley.com/doi/https://doi.org/10.1111/j.1365-2796.2010.02274.x
19. Chang, C., Deng, Y., Jiang, X. & Long, Q. Multiple imputation for analysis of incomplete data in distributed health data networks. *Nat. Commun.* **11**, 5467 (2020).
20. Li, Y. et al. BEHRT: transformer for electronic health records. *Sci. Rep.* **10**, 7155 (2020).
21. Li, Y. et al. Hi-BEHRT: hierarchical Transformer-Based model for accurate prediction of clinical events using multimodal longitudinal electronic health records. *IEEE J. Biomedical Health Inf.* **27**, 1106–1117 (2023).
22. Zhan, X., Humbert-Droz, M., Mukherjee, P. & Gevaert, O. Structuring clinical text with AI: old versus new natural Language processing techniques evaluated on eight common cardiovascular diseases. *Patterns* **2**, 100289 (2021).
23. Zou, Y. et al. Modeling electronic health record data using an end-to-end knowledge-graph-informed topic model. *Sci. Rep.* **12**, 17868 (2022).
24. Du, Y., Rafferty, A. R., McAuliffe, F. M., Wei, L. & Mooney, C. An explainable machine learning-based clinical decision support system for prediction of gestational diabetes mellitus. *Sci. Rep.* **12**, 1170 (2022).
25. Shishegar, R. et al. Using imputation to provide harmonized longitudinal measures of cognition across AIBL and ADNI. *Sci. Rep.* **11**, 23788 (2021).
26. Azur, M. J., Stuart, E. A., Frangakis, C. & Leaf, P. J. Multiple imputation by chained equations: what is it and how does it work? *Int. J. Methods Psychiatr Res.* **20**, 40–49 (2011).
27. Stekhoven, D. J. & Bühlmann, P. MissForest—non-parametric missing value imputation for mixed-type data. *Bioinformatics* **28**, 112–118 (2012).
28. Wolpert, D. H. & Macready, W. G. No free lunch theorems for optimization. *IEEE Trans. Evol. Comput.* **1**, 67–82 (1997).
29. He, K., Zhang, X., Ren, S. & Sun, J. Delving deep into rectifiers: surpassing Human-Level performance on imagenet classification. in 1026–1034 (2015).
30. Breiman, L. Random forests. *Mach. Learn.* **45**, 5–32 (2001).
31. LIBSVM. A library for support vector machines: ACM Transactions on Intelligent Systems and Technology: Vol 2, No 3. https://doi.org/10.1145/1961189.1961199
32. Chen, T., Guestrin, C. & XGBoost: A Scalable Tree Boosting System. in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* 785–794Association for Computing Machinery, New York, NY, USA, (2016). https://doi.org/10.1145/2939672.2939785
33. Johnson, A. E. W. et al. MIMIC-III, a freely accessible critical care database. *Sci. Data.* **3**, 160035 (2016).
34. Fleurence, R. L. et al. Launching PCORnet, a National patient-centered clinical research network. *J. Am. Med. Inform. Assoc.* **21**, 578–582 (2014).
35. Batista, G. E. A. P. A. & Monard, M. C. An analysis of four missing data treatment methods for supervised learning. *Appl. Artif. Intell.* **17**, 519–533 (2003).
36. von Hippel, P. T. Should a normal imputation model be modified to impute skewed variables?? *Sociol. Methods Res.* **42**, 105–138 (2013).
37. Shang, C. et al. VIGAN: Missing view imputation with generative adversarial networks. in. *IEEE International Conference on Big Data (Big Data)* 766–775 (2017). (2017). https://doi.org/10.1109/BigData.2017.8257992
38. Yoon, J., Jordon, J. & Schaar, M. G. A. I. N. Missing Data Imputation using Generative Adversarial Nets. in *Proceedings of the 35th International Conference on Machine Learning* 5689–5698PMLR, (2018).
39. Nazábal, A., Olmos, P. M., Ghahramani, Z. & Valera, I. Handling incomplete heterogeneous data using VAEs. *Pattern Recogn.* **107**, 107501 (2020).
40. Genomic data imputation. with variational auto-encoders | GigaScience | Oxford Academic. https://academic.oup.com/gigascience/article/9/8/giaa082/5881619
41. Wang, Y., Li, D., Li, X. & Yang, M. PC-GAIN: Pseudo-label conditional generative adversarial imputation networks for incomplete data. *Neural Netw.* **141**, 395–403 (2021).
42. Biessmann, F., Salinas, D., Schelter, S., Schmidt, P. & Lange, D. 'Deep' Learning for Missing Value Imputationin Tables with Non-Numerical Data. in *Proceedings of the 27th ACM International Conference on Information and Knowledge Management* 2017–2025Association for Computing Machinery, New York, NY, USA, (2018). https://doi.org/10.1145/3269206.3272005
43. Jäger, S., Allhorn, A. & Bießmann, F. A benchmark for data imputation methods. *Frontiers Big Data* **4**, (2021).
44. Borisov, V. et al. Deep neural networks and tabular data: A survey. *IEEE Trans. Neural Networks Learn. Syst.* **1–21** https://doi.org/10.1109/TNNLS.2022.3229161 (2022).
45. Shwartz-Ziv, R. & Armon, A. Tabular data: deep learning is not all you need. *Inform. Fusion.* **81**, 84–90 (2022).
46. Grinsztajn, L., Oyallon, E. & Varoquaux, G. Why do tree-based models still outperform deep learning on tabular data? Preprint at (2022). https://doi.org/10.48550/arXiv.2207.08815
47. Khademi, A. Flexible Imputation of Missing Data (2nd Edition). *Journal of Statistical Software* 93, 1–4 (2020).
48. Rigby, R. A. & Stasinopoulos, D. M. Generalized additive models for location, scale and shape. *J. Royal Stat. Soc. Ser. C: Appl. Stat.* **54**, 507–554 (2005).
49. Flexible Regression and Smoothing. Using GAMLSS in R. *Routledge & CRC Press* https://www.routledge.com/Flexible-Regression-and-Smoothing-Using-GAMLSS-in-R/Stasinopoulos-Rigby-Heller-Voudouris-Bastiani/p/book/9780367658069
50. Salfran, D. & Spiess, M. Generalized additive model multiple imputation by chained equations with package ImputeRobust. *R J.* **10**, 61 (2018).
51. Williamson, B. D. & Huang, Y. Flexible variable selection in the presence of missing data. *Int J. Biostat* **20**, 347–359 .
52. Kramer, O. Scikit-Learn. in Machine Learning for Evolution Strategies (ed Kramer, O.) 45–53 (Springer International Publishing, Cham, doi:https://doi.org/10.1007/978-3-319-33383-0_5. (2016).
53. Rubin, D. B. Multiple imputation. in *Flexible Imputation of Missing Data, Second Edition* (Chapman and Hall/CRC, (2018).

Acknowledgements

Research reported in this publication was supported in part by the National Institute Of Neurological Disorders And Stroke of the National Institutes of Health under Award Number R01NS128986. This study is also funded by the National Institute of Health grants No. 5U01DK127384-04, and 3U01DK127384-03. The content is solely the responsibility of the authors and does not necessarily represent the official views of the National Institutes of

Health. Part of software development and numerical computing was carried out on Baobab and Yggdrasil at the computing cluster of the University of Geneva.

Author contributions
VA and RZ conceptualized the study, supervised the project, and secured funding for its execution. AVS developed the imputation software, conducted the calculations for the figures, and was responsible for the preparation of MIMIC and Penn State data and wrote the original draft. JL, performed analysis on the Geisinger data, and prepared the simulated data. WH assisted with the Penn State data collection efforts. VA, RZ, WH, MY, MW, and HL significantly contributed by thoroughly reviewing and editing the manuscript for intellectual content and interpretation of findings.

Declarations

Competing interests
The authors declare no competing interests.

Supplementary Information
Supplementary material is available online.

MIMIC data
The MIMIC dataset, version 1.4, utilized in this study is publicly available and can be accessed online[6].

Simulated Data
The simulated dataset, which emulates the missing values observed in the real-world Geisinger data, will be made available upon request, request can be made to Jiang Li (jli@geisinger.edu). The simulated data is accessible via the GitHub repository at https://github.com/TheDecodeLab/simulation-of-laboratory-variables-using-Multivariate-Gaussian-approach.

Geisinger data
Due to privacy and confidentiality concerns, we are unable to publicly share the Geisinger real-world dataset. The data was collected from a large integrated healthcare system encompassing multiple hospitals in the United States, necessitating strict adherence to privacy regulations and ethical considerations. However, data can be shared upon execution of a data-sharing agreement; interested parties can contact Jiang Li (jli@geisinger.edu) to request access.

Penn state health data
Due to privacy and confidentiality concerns, we are also unable to publicly share the Penn State Health real-world dataset. However, data can be shared upon execution of a data-sharing agreement; interested parties can contact Vida Abedi or Wenke Hwang (vabedi@pennstatehealth.psu.edu or whwang@pennstatehealth.psu.edu) to request access.

Scripts and *Pympute* Package
To promote transparency and enable reproducibility, the code used in this research and the *Pympute* package is released as open-source resources on GitHub (https://github.com/TheDecodeLab) (upon publication). Interested researchers can access the codebase, utilize *Pympute* for their imputation tasks, and replicate the methodology employed in this study.

Additional information
Supplementary Information The online version contains supplementary material available at https://doi.org/10.1038/s41598-025-02276-5.

Correspondence and requests for materials should be addressed to V.A.

Reprints and permissions information is available at www.nature.com/reprints.

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.