Closing Gaps: An Imputation Analysis of ICU Vital Signs

Robin van de Water[1], **Bert Arnrich**[1]
[1] Hasso Plattner Institute, University of Potsdam, Germany
{robin.vandewater, bert.arnrich}@hpi.de

Abstract

As more ICU EHR data becomes available, the interest in developing clinical prediction models to improve healthcare protocols increases. However, lacking data quality still hinders clinical prediction using Machine Learning (ML). Many vital sign measurements, such as heart rate, contain sizeable missing segments, leaving gaps in the data that could negatively impact prediction performance. Previous works have introduced numerous time-series imputation techniques. Nevertheless, more comprehensive work is needed to compare a representative set of methods for imputing ICU vital signs to determine the best practice. In reality, ad-hoc imputation techniques that could decrease prediction accuracy, like zero imputation, are still used. In this work, we compare established imputation techniques to guide researchers in improving clinical prediction model performance by choosing the most accurate imputation technique. We introduce an extensible, reusable benchmark with, currently, 15 imputation and 4 amputation methods created for benchmarking on major ICU datasets. We hope to provide a comparative basis and facilitate further ML development to bring more models into clinical practice.

Software Repository: `https://github.com/rvandewater/YAIB`

1 Introduction

Real-world environments like medical treatment centers often collect large amounts of data through administration, tests, and monitoring equipment. Although patients get monitored frequently during their ICU stay, many practical issues can lead to data loss and missingness; data quality is an ongoing issue for clinical prediction modeling. Some of this is by design (an infrequent sample rate due to staff shortage) or unintentionally (monitoring devices lose connection). Clinical prediction is a field that emerged from Electronic Health Record (EHR) data that was formerly used for administration. For example, we could predict mortality [2] or sepsis [26] within the ICU. In order to utilize EHRs to create useful clinical models, the standard practice in ML is to use imputation methods, which are algorithms to fill in missing data. Similarly to the increase of advanced methods in ML and Deep Learning (DL) in recent years, the field of imputation has also rapidly developed, leaving us with the question of which technique to use [50]. Additionally, we note that there could be a difference of the performance of imputation methods when considering downstream tasks like the aforementioned sepsis and mortality [36, 51].

Our work analyses three large open-access ICU datasets with four types and three quantities of missingness in vital signs. In total, we benchmark 15 different models for imputation. This selection includes numerous techniques, including recently introduced generative models (Diffusion) and attention-based techniques. The methods are incorporated within a benchmarking framework to allow for the replication of our experiments and reuse on current and future datasets. This setup allows researchers to build upon our results and create, benchmark, and compare imputation methods on

1st Workshop on Deep Generative Models for Health at NeurIPS 2023.

standard ICU datasets. Lastly, we provide a straightforward experiment pipeline to apply imputation to any clinical dataset of choice.

2 Background & Related Work

2.1 Types of missingness

We define *missingness* as the absence of data where it is unexpected. For example, an absence of data is expected when we consider time-series observations below the technical sampling frequency of a vital sign recording device. We identify four significant types of missingness in this work: *missing completely at random (MCAR)*, *missing at random (MAR)*, *missing not at random (MNAR)*, and *Blackout (BO)*. Below, each method and the implementation in our framework to simulate this type of missingness is detailed; we call the latter the *amputation* mechanism as it removes data to assess imputation performance.

MCAR refers to where missingness introduces no statistical bias. For example, in a clinical trial, a recording of a group of patients might be missing due to failing equipment. **MAR** does introduce a bias, but this bias is systematically related exclusively to observed data. For example, we record the sex as a response to a survey, and men are less likely to respond. With **MNAR**, data are missing, that are systematically related to unobserved factors (i.e., events not measured in the experiment). We again take the survey example but assume that sex is not recorded in this case; this gives us a scenario where it is hard to account for bias introduced by the missingness. For **BO** an entire subset of data is missing for several features during several timesteps and dimensions (or features). This type of missingness is less commonly tested but has a basis in existing literature [1].

2.2 Datasets

We have used three major ICU datasets in our work provided in the Yet Another ICU Benchmark (YAIB) [45] experiment framework. The framework already implements a flexible framework for downstream tasks using most open access ICU datasets, thus, we judge it a good basis to further expand. The MIMIC-III dataset [21] is the most commonly used ML prediction [42]. The newer MIMIC-IV (MIIV) includes several improvements, including more and newer patient records and a revised structure including regular hospital information; we use this version for our experiments. The eICU Collaborative Research Database (eICU) [31] is the first sizable multi-center dataset. The High tIme Resolution ICU dataset (HiRID) was collected at Bern University Hospital, Switzerland, and has incorporated more observations than the other datasets [15]. A more comprehensive overview can be found in Table 3 and Sauer et al. [35].

Choice of Features For our comparison of imputation methods, we chose six temporal vital signs that showed significantly lower missingness than other recorded variables for each dataset: *heart rate (hr), respiratory rate (resp), oxygen saturation (o2sat), mean arterial pressure (map), systolic blood pressure (sbp), and diastolic blood pressure (dbp)*. The nature of ICU data recording likely causes this pattern: monitoring equipment is usually continuously attached and allows for non-invasive recording, whereas, for example, lab values are taken at most several times per day and involve manual labor. We select these variables from the 52 (4 static, 48 temporal) variables in the harmonized datasets (downsampled to hour [45]) to ensure we have enough ground truth data to assess any imputation method's performance accurately. Figure 4 show missingness correlation between these features; Figure 5 aims to explore informative missingness by comparing the population that died within the ICU with those who survived.

2.3 Imputation methods

We have conducted a systematic literature review to discover promising imputation technologies. Table 1 shows the imputation methods we benchmark in this work. We distinguish several categories, as shown on the left side of the table. **Baselines** are still commonly used in many applications of data preparation of ML modeling as they are deterministic and computationally cheap. **Algorithmic** methods [4, 40], use statistical assumptions on the data to create iterative algorithms; the methods are robust for simpler data. We include several **Deep Learning (DL)** methods, including a simple

TABLE 1: Overview of the implemented imputation methods.

		Abbreviation	Original Publication	Year	Source	Novelty
Naive		Zero	-		ω [28][1]	Baseline
		Median	-		ω [28][1]	Baseline
		Mean	-		ω [28][1]	Baseline
		MostFrequent	-		ω [28][1]	Baseline
Algo.		MICE	Buuren et al. [4]	2011	ω [18][2]	Equation based benchmark
		MissForest	Stekhoven et al. [40]	2012	ω [18][2]	Random forest based
Deep learning		MLP	Junninen et al. [22]	2004	σ	DL baseline
	RNN	BRITS	Cao et al. [5]	2018	ω [9][4]	Bidirectional RNN
		GRU-D	Che et al. [6]	2018	α [7][3]	Bidirectional LSTM/GRU
		M4IP	Shi et al. [38]	2021	α [7][3]	State decay RGRU-D
	Att.	Attention	Vaswani et al. [46]	2017	ω [9][4]	Attention-based approach
		Neural Processes	Garnelo et al. [12]	2018	σ	First Neural Processes
		SAITS	Du et al. [10]	2022	ω [9][4]	SOTA Attention-based
	Gen.	Diffusion	Ho et al. [14]	2020	σ [27][5]	Prob. Diff. model
		CSDI	Tashiro et al. [44]	2022	α [44][6]	Conditional Diff. model

ω Wrapper framework α Adapted using open-acces code σ Self-implemented based on paper description [1] `https://github.com/scikit-learn/scikit-learn` [2] `https://github.com/vanderschaarlab/hyperimpute` [3] `https://github.com/Graph-Machine-Learning-Group/grin` [4]`https://github.com/WenjieDu/PyPOTS` [5] `https://github.com/diff-usion/Awesome-Diffusion-Models` [6] `https://github.com/ermongroup/CSDI`

Multilayer Perceptron. **RNN**-based methods [5, 6, 38] have traditionally performed well on time-series prediction and imputation. **Attention** methods [10, 12, 46] have developed rapidly in the past years and shown potential for various prediction tasks. **Generative** include the more recent diffusion models [14, 44]. Although these methods are more complex than previous methods, they have significantly impacted the field of DL in recent years.

Medical time series imputation benchmarks We recognize several earlier attempts at collecting and benchmarking imputation methods [16, 18, 24, 29, 32, 36, 41] (Table 4). Jäger et al. [16] investigated the performance of six imputation methods; however, there was no medical time-series among the tested datasets. Perez-Lebel et al. [30] focused on ML and algorithmic methods of imputation; they do not include any DL imputation methods in the analysis. Psychogyios et al. [32] conducted a benchmark of imputation methods on a closed-source tabular dataset. Luo [24] reports on a challenge; the investigated dataset and methods limit the applicability to current data. Jarrett et al. [18] introduces the HyperImpute framework; we provide a more comprehensive medical vital sign analysis [11]. Sun et al. [41] investigates nine imputation methods on medical datasets. However, they do not provide an open-access extensible framework for implementing the imputation methods in a downstream task. Finally, Shadbahr et al. [36] investigates five imputation methods; the choice of methods and datasets is limited compared to our approach.

Our work uses a variety of missingness patterns. It allows for evaluating both the imputation and downstream classification task, ensuring that the benchmarks indicate real-world performance. Additionally, we utilize three ICU datasets and provide a transparent, publicly available, experimental setup, ensuring reproducibility and enabling easier comparison of results. Future work includes more imputation methods (>25 in total, see Table 4) results once we have verified their implementation.

3 Comparing imputation technique performance

We provide a standardized interface for imputation methods. We have utilized this to **1)** wrap interfaces of earlier frameworks [18, 28], **2)** adapt open-source code to our interface [1, 7, 44], and **3)** create imputation methods without an existing code-base. We use the Hyperimpute [18] and PyPOTS [9] frameworks as we judged them to be stable and flexible enough to use. The rest of the methods are implemented directly in YAIB [45]. Hyperparameter tuning was performed for the DL and ML methods. Both the methods from the imputation packages and the methods implemented within YAIB are wrapped in an `ImputationWrapper` interface, which derives from a Pytorch-lightning `DLWrapper` module (see Appendix D to implement new imputation methods). Additionally, we developed the `ampute_data(missing_type, missing_amount)` functionality which generates the dataset with artificially introduced missing values and a boolean mask indicating their location. The code is provided in the appendix. Using this function, one can quickly generate datasets with

different types and levels of missingness to test and evaluate current and future imputation methods. We shortly describe the implementation of each missingness technique.

The **MCAR** amputation mechanism generates missing values randomly without considering any additional input from the data or its characteristics. For the **MAR** amputation mechanism, we select a subset of fully observed variables. Then, missing values are introduced to the remaining variables by a logistic model [25]. The proportion of missing values in these variables is re-scaled to match the desired proportion of overall missingness. The **MNAR** also utilizes a logistic masking model. We split the variables into a set of inputs for a logistic model and a set whose missing probabilities are determined by the logistic model. The coefficients for the logistic masking model are selected such that $W^\top x$ has a unit variance, where W is the subset of observed variables, and x is the corresponding missing variable [39]. Then, inputs are masked; the missing values from the second set will depend on masked values. Finally, the **BO** implementation takes a proportion of missing values; the function randomly selects rows in the data matrix and sets all their values to missing.

We have chosen *Mean Absolute Error (MAE)*, *Root Mean Square Error (RMSE)*, and *Jensen-Shannon Divergence (JSD)* (lower is better for each) as evaluation metrics to show 1) the averaged error over every time-series 2) penalize higher individual errors more strongly, and 3) demonstrate if an imputation method is capable of reconstructing the original distribution.

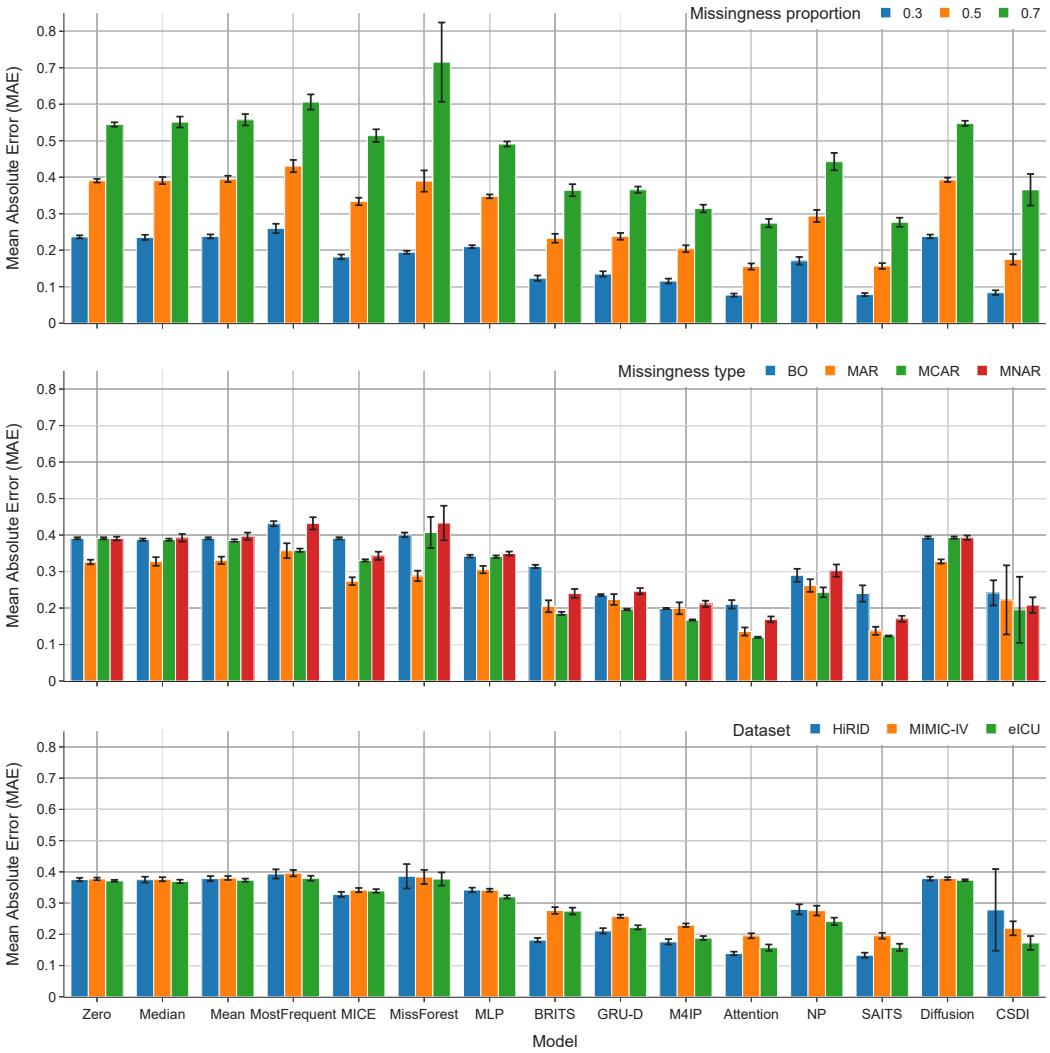


FIGURE 1: *Performance in MAE across the selected imputation methods in three dimensions.* **Top:** imputation methods separated by missingness proportion for MNAR. **Middle:** aggregated performance per missingness type. **Bottom:** aggregated performance for each dataset.

TABLE 2: *The three best RNN, Attention, and Generative type imputation methods.* We **embolden** the best method per missingness type and metric including those within a standard deviation (±).

Missingness	Metric	GRU-D	Attention	CSDI
MCAR	RMSE	5±0	374±4	**4±0**
	MAE	0.18±0.00	**0.14±0.00**	0.15±0.00
	JSD	**0.02±0.00**	**0.02±0.00**	**0.02±0.00**
MAR	RMSE	**5±0**	372±32	6±4
	MAE	0.19±0.02	**0.13±0.01**	0.18±0.08
	JSD	0.04±0.01	**0.02±0.00**	0.03±0.01
MNAR	RMSE	**5±0**	419±27	**5±0**
	MAE	0.20±0.01	**0.16±0.01**	**0.17±0.01**
	JSD	0.03±0.00	**0.02±0.00**	**0.02±0.00**
BO	RMSE	**5±0**	485±30	**5±1**
	MAE	**0.18±0.00**	0.20±0.02	0.19±0.00
	JSD	0.03±0.00	0.03±0.01	**0.02±0.00**

3.1 Results

We performed experiments with the introduced amputation methods, datasets, and imputation methods (Table 1). Figure 1 presents the results per imputation mechanism and the amount of missingness for MAE. Results for RMSE (Figure 2) and JSD (Figure 3) can be found in the appendix.

The top graph compares imputation methods for missing not at random, a realistic missingness type for ICU data. This plot shows that, as expected, higher missingness is harder to impute and results in higher standard deviations. 30% missingness has at least half the MAE of 70%. In the middle, we can conclude that BO and MNAR are generally the hardest to impute. Moreover, CSDI has a high variance, which might indicate more iterations are needed for these experiments. The below plot shows HiRID is generally the easiest to impute, followed by eICU. In these plots we see that Attention imputation slightly bests the other methods for MAE in each of the dimensions (missingness proportion, missingness type, and dataset).

Curiously, we do not observe a decisive trend which indicates that more recent models perform better; a relatively older model, GRU-D, seems to compete with more sophisticated models. Table 2, further describes the performance of the three best imputation techniques per category for each type of missingness and performance metric. The results are comparable for MAE and JSD, where the three methods are comparable; attention slightly besting the other methods. When it comes to RMSE, however, GRU-D and CSDI demonstrate significantly better performance; this could indicate that Attention imputation, along with several other methods (Figure 2), has a large error for the value of individual predictions. If we require an algorithmic method, for example, we have no GPUs or explainability is required, MICE seems to be the best choice. Lastly, naive methods have comparable performance although median has the best performance across metrics.

4 Discussion

Whereas newer methods often promise SOTA performance, the results depend on the type of task, and benchmarking may involve cherry-picking. RNN-type, attention, and generative models show promise for imputing time-series vital signs. The best technique depends on the type of missingness, the percentage of missingness, and the desired metric to minimize. We provide an openly accessible, extensible, testbed to compare current and future imputation techniques on a medical dataset set of choice using YAIB.

To make our comparison more robust, future work aims to include more diverse features, datasets, types of imputation methods. Additionally, we recognize the importance of including downstream task performance as well as a more thorough discussion on the clinical applicability of imputation methods in terms. Our aim is to work towards a decision guide for machine learning in health that can be used by clinicians and ML-researchers and increase common understanding.

Acknowledgments

Robin van de Water is funded by the is funded by the "Gemeinsamer Bundesausschuss (G-BA) Innovationsausschuss" in the framework of "CASSANDRA - Clinical ASSist AND aleRt Algorithms" (project number 01VSF20015). We would like to acknowledge the work of Alisher Turubayev, Anna Shopova, Fabian Lange, Mahmut Kamalak, Paul Mattes, and Victoria Ayvasky for adding several imputation methods to the repository.

References

[1] Juan Miguel Lopez Alcaraz and Nils Strodthoff. *Diffusion-Based Time Series Imputation and Forecasting with Structured State Space Models*. Feb. 2023. arXiv: 2208.09399 [cs, stat].

[2] Stephanie Baker, Wei Xiang, and Ian Atkinson. "Continuous and Automatic Mortality Risk Prediction Using Vital Signs in the Intensive Care Unit: A Hybrid Neural Network Approach". In: *Scientific Reports* 10.1 (Dec. 2020), p. 21282. ISSN: 2045-2322. DOI: 10.1038/s41598-020-78184-7.

[3] Sebastiano Barbieri et al. "Benchmarking Deep Learning Architectures for Predicting Readmission to the ICU and Describing Patients-at-Risk". In: *Scientific Reports* 10.1 (Jan. 2020), p. 1111. ISSN: 2045-2322. DOI: 10.1038/s41598-020-58053-z.

[4] Stef van Buuren and Karin Groothuis-Oudshoorn. "Mice: Multivariate Imputation by Chained Equations in R". In: *Journal of Statistical Software* 45 (Dec. 2011), pp. 1–67. ISSN: 1548-7660. DOI: 10.18637/jss.v045.i03.

[5] Wei Cao et al. *BRITS: Bidirectional Recurrent Imputation for Time Series*. May 2018. DOI: 10.48550/arXiv.1805.10572. arXiv: 1805.10572 [cs, stat].

[6] Zhengping Che et al. "Recurrent Neural Networks for Multivariate Time Series with Missing Values". In: *Scientific Reports* 8.1 (Dec. 2018), p. 6085. ISSN: 2045-2322. DOI: 10.1038/s41598-018-24271-9.

[7] Andrea Cini, Ivan Marisca, and Cesare Alippi. *Filling the G_ap_s: Multivariate Time Series Imputation by Graph Neural Networks*. Feb. 2022. DOI: 10.48550/arXiv.2108.00298. arXiv: 2108.00298 [cs].

[8] Wenjie Du. *PyPOTS: A Python Toolbox for Data Mining on Partially-Observed Time Series*. May 2023. DOI: 10.48550/arXiv.2305.18811. arXiv: 2305.18811 [cs, stat].

[9] Wenjie Du. "PyPOTS: A Python Toolbox for Machine Learning on Partially-Observed Time Series". In: *9th SIGKDD Workshop on Mining and Learning from Time Series (MiLeTS'23)*. 2023.

[10] Wenjie Du, David Cote, and Yan Liu. *SAITS: Self-Attention-based Imputation for Time Series*. May 2022. arXiv: 2202.08516 [cs].

[11] Dheeru Dua and Casey Graff. *UCI Machine Learning Repository*. 2017.

[12] Marta Garnelo et al. "Neural Processes". In: *arXiv preprint arXiv:1807.01622* (2018). arXiv: 1807.01622.

[13] Hrayr Harutyunyan et al. "Multitask Learning and Benchmarking with Clinical Time Series Data". In: *Scientific Data* 6.1 (Dec. 2019), p. 96. ISSN: 2052-4463. DOI: 10.1038/s41597-019-0103-9.

[14] Jonathan Ho, Ajay Jain, and Pieter Abbeel. "Denoising Diffusion Probabilistic Models". In: *Advances in Neural Information Processing Systems*. Vol. 33. Curran Associates, Inc., 2020, pp. 6840–6851.

[15] Stephanie L Hyland. "Early Prediction of Circulatory Failure in the Intensive Care Unit Using Machine Learning". In: *Nature MedIcIne* 26 (2020), p. 28.

[16] Sebastian Jäger, Arndt Allhorn, and Felix Bießmann. "A Benchmark for Data Imputation Methods". In: *Frontiers in Big Data* 4 (July 2021), p. 693674. ISSN: 2624-909X. DOI: 10.3389/fdata.2021.693674.

[17] Daniel Jarrett et al. "CLAIRVOYANCE: A PIPELINE TOOLKIT FOR MEDICAL TIME SERIES". In: *International Conference on Learning Representations* (2021), p. 32.

[18] Daniel Jarrett et al. *HyperImpute: Generalized Iterative Imputation with Automatic Model Selection*. June 2022. DOI: 10.48550/arXiv.2206.07769. arXiv: 2206.07769 [cs, stat].

[19] Alistair E. W. Johnson, Tom J. Pollard, and Roger G. Mark. "Reproducibility in Critical Care: A Mortality Prediction Case Study". In: *Proceedings of the 2nd Machine Learning for Healthcare Conference*. PMLR, Nov. 2017, pp. 361–376.

[20] Alistair E. W. Johnson et al. "MIMIC-IV, a Freely Accessible Electronic Health Record Dataset". In: *Scientific Data* 10.1 (Jan. 2023), p. 1. ISSN: 2052-4463. DOI: 10.1038/s41597-022-01899-x.

[21] Alistair E.W. Johnson et al. "MIMIC-III, a Freely Accessible Critical Care Database". In: *Scientific Data* 3.1 (Dec. 2016), p. 160035. ISSN: 2052-4463. DOI: 10.1038/sdata.2016.35.

[22] Heikki Junninen et al. "Methods for Imputation of Missing Values in Air Quality Data Sets". In: *Atmospheric Environment* 38.18 (June 2004), pp. 2895–2907. ISSN: 13522310. DOI: 10.1016/j.atmosenv.2004.02.026.

[23] Zhifeng Kong et al. *DiffWave: A Versatile Diffusion Model for Audio Synthesis*. Mar. 2021. arXiv: 2009.09761 [cs, eess, stat].

[24] Yuan Luo. "Evaluating the State of the Art in Missing Data Imputation for Clinical Data". In: *Briefings in Bioinformatics* 23.1 (Jan. 2022), pp. 1–9. ISSN: 1467-5463, 1477-4054. DOI: 10.1093/bib/bbab489.

[25] Imke Mayer et al. *R-Miss-Tastic: A Unified Platform for Missing Values Methods and Workflows*. Aug. 2021. DOI: 10.48550/arXiv.1908.04822. arXiv: 1908.04822 [stat].

[26] Michael Moor et al. "Early Recognition of Sepsis with Gaussian Process Temporal Convolutional Networks and Dynamic Time Warping". In: *Proceedings of the 4th Machine Learning for Healthcare Conference*. PMLR, Oct. 2019, pp. 2–26.

[27] Alex Nichol and Prafulla Dhariwal. *Improved Denoising Diffusion Probabilistic Models*. Feb. 2021. DOI: 10.48550/arXiv.2102.09672. arXiv: 2102.09672 [cs, stat].

[28] F. Pedregosa et al. "Scikit-Learn: Machine Learning in Python". In: *Journal of Machine Learning Research* 12 (2011), pp. 2825–2830.

[29] Alexandre Perez-Lebel et al. *Benchmarking Missing-Values Approaches for Predictive Models on Health Databases*. Feb. 2022. DOI: 10.48550/arXiv.2202.10580. arXiv: 2202.10580 [cs].

[30] Alexandre Perez-Lebel et al. "Benchmarking Missing-Values Approaches for Predictive Models on Health Databases". In: *GigaScience* 11 (Apr. 2022), giac013. ISSN: 2047-217X. DOI: 10.1093/gigascience/giac013.

[31] Tom J. Pollard et al. "The eICU Collaborative Research Database, a Freely Available Multi-Center Database for Critical Care Research". In: *Scientific Data* 5.1 (Dec. 2018), p. 180178. ISSN: 2052-4463. DOI: 10.1038/sdata.2018.178.

[32] Konstantinos Psychogyios et al. "Missing Value Imputation Methods for Electronic Health Records". In: *IEEE Access* 11 (2023), pp. 21562–21574. ISSN: 2169-3536. DOI: 10.1109/ACCESS.2023.3251919.

[33] Sanjay Purushotham et al. "Benchmarking Deep Learning Models on Large Healthcare Datasets". In: *Journal of Biomedical Informatics* 83 (July 2018), pp. 112–134. ISSN: 15320464. DOI: 10.1016/j.jbi.2018.04.007.

[34] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. "U-Net: Convolutional Networks for Biomedical Image Segmentation". In: *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*. Ed. by Nassir Navab et al. Lecture Notes in Computer Science. Cham: Springer International Publishing, 2015, pp. 234–241. ISBN: 978-3-319-24574-4. DOI: 10.1007/978-3-319-24574-4_28.

[35] Christopher M. Sauer et al. "Systematic Review and Comparison of Publicly Available ICU Data Sets—A Decision Guide for Clinicians and Data Scientists". In: *Critical Care Medicine* 50.6 (June 2022), e581–e588. ISSN: 0090-3493. DOI: 10.1097/CCM.0000000000005517.

[36] Tolou Shadbahr et al. *Classification of Datasets with Imputed Missing Values: Does Imputation Quality Matter?* June 2022. DOI: 10.48550/arXiv.2206.08478. arXiv: 2206.08478 [cs].

[37] Seyedmostafa Sheikhalishahi, Vevake Balaraman, and Venet Osmani. "Benchmarking Machine Learning Models on Multi-Centre eICU Critical Care Dataset". In: *PLOS ONE* 15.7 (July 2020). Ed. by Kyoung-Sae Na, e0235424. ISSN: 1932-6203. DOI: 10.1371/journal.pone.0235424.

[38] Zhenkun Shi et al. "Deep Dynamic Imputation of Clinical Time Series for Mortality Prediction". In: *Information Sciences* 579 (Nov. 2021), pp. 607–622. ISSN: 0020-0255. DOI: 10.1016/j.ins.2021.08.016.

[39] Aude Sportisse. "Sp´ecialit´e Doctorale: Statistique". In: ().

[40] Daniel J. Stekhoven and Peter Bühlmann. "MissForest—Non-Parametric Missing Value Imputation for Mixed-Type Data". In: *Bioinformatics* 28.1 (Jan. 2012), pp. 112–118. ISSN: 1367-4803. DOI: 10.1093/bioinformatics/btr597.

[41] Chenxi Sun et al. *A Review of Deep Learning Methods for Irregularly Sampled Medical Time Series Data*. Oct. 2020. DOI: 10.48550/arXiv.2010.12493. arXiv: 2010.12493 [cs, stat].

[42] Mahanazuddin Syed et al. "Application of Machine Learning in Intensive Care Unit (ICU) Settings Using MIMIC Dataset: Systematic Review". In: *Informatics* 8.1 (Mar. 2021), p. 16. ISSN: 2227-9709. DOI: 10.3390/informatics8010016.

[43] Shengpu Tang et al. "Democratizing EHR Analyses with FIDDLE: A Flexible Data-Driven Preprocessing Pipeline for Structured Clinical Data". In: *Journal of the American Medical Informatics Association* 27.12 (Dec. 2020), pp. 1921–1934. ISSN: 1527-974X. DOI: 10.1093/jamia/ocaa139.

[44] Yusuke Tashiro et al. *CSDI: Conditional Score-based Diffusion Models for Probabilistic Time Series Imputation*. Oct. 2021. DOI: 10.48550/arXiv.2107.03502. arXiv: 2107.03502 [cs, stat].

[45] Robin van de Water et al. *Yet Another ICU Benchmark: A Flexible Multi-Center Framework for Clinical ML*. June 2023. arXiv: 2306.05109 [cs].

[46] Ashish Vaswani et al. "Attention Is All You Need". In: *arXiv:1706.03762 [cs]* (Dec. 2017). arXiv: 1706.03762 [cs].

[47] Shirly Wang et al. "MIMIC-Extract: A Data Extraction, Preprocessing, and Representation Pipeline for MIMIC-III". In: *Proceedings of the ACM Conference on Health, Inference, and Learning*. Toronto Ontario Canada: ACM, Apr. 2020, pp. 222–235. ISBN: 978-1-4503-7046-2. DOI: 10.1145/3368555.3384469.

[48] Feng Xie et al. "Benchmarking Emergency Department Prediction Models with Machine Learning and Public Electronic Health Records". In: *Scientific Data* 9.1 (Oct. 2022), p. 658. ISSN: 2052-4463. DOI: 10.1038/s41597-022-01782-9.

[49] Hugo Yèche et al. "HiRID-ICU-Benchmark – A Comprehensive Machine Learning Benchmark on High-resolution ICU Data". In: *arXiv:2111.08536 [cs]* (Jan. 2022). arXiv: 2111.08536 [cs].

[50] Aznilinda Zainuddin et al. "Time Series Data and Recent Imputation Techniques for Missing Data: A Review". In: *2022 International Conference on Green Energy, Computing and Sustainable Technology (GECOST)*. Oct. 2022, pp. 346–350. DOI: 10.1109/GECOST55694.2022.10010499.

[51] Zhihui Zhang et al. "False Positive Findings during Genome-Wide Association Studies with Imputation: Influence of Allele Frequency and Imputation Accuracy". In: *Human Molecular Genetics* 31.1 (Aug. 2021), pp. 146–155. ISSN: 0964-6906. DOI: 10.1093/hmg/ddab203.

A Background and related work

TABLE 3: *Supplemental details of openly accessible ICU datasets.* Note that accessing each dataset requires completing a credentialing procedure.

Dataset	**MIMIC-III / IV**	**eICU**	**HiRID**
Stays*	40k / 73k	201k	34k
Version	v1.4 / v2.2	v2.0	v1.1.1
Frequency	1 hour	5 minutes	2 / 5 minutes
Origin	USA	USA	Switzerland
Published	2015 [21] / 2020 [20]	2017 [31]	2020 [15]
Benchmark	[3, 13, 17, 19, 33, 43, 47]/ [48]	[37, 43]	[49]
Repository link	Physionet/ Physionet	Physionet	Physionet

TABLE 4: *Imputation benchmarks for medical time series.*

		Jäger et al. [16]	Sun et al. [41]	Perez-Lebel et al. [29]	Shadbahr et al. [36]	Psychogyios et al. [32]	Luo [24]	Jarrett et al. [18]	**Our Work**
Task	Imputation	✓	✓	✗	✓	✓	✓	✓	✓
	Downstream Task	✓	✓	✓	✓	✓	✗	✗	✓
Methods	Naive	✓	✗	✓	✓	✓	✓	✓	✓
	Algorithmic	✗	✗	✓	✓	✗	✓	✓	✓
	Machine Learning	✓	✗	✓	✓	✓	✓	✓	✓
	GAN-based	✓	✓	✗	✓	✓	✗	✓	✓
	RNN-based	✗	✓	✗	✗	✗	✓	✗	✓
	AE-based	✓	✗	✗	✓	✓	✗	✓	✓
	Attention-based	✗	✗	✗	✗	✗	✗	✗	✓
	Diffusion Models	✗	✗	✗	✗	✗	✗	✗	✓
	Neural Processes	✗	✗	✗	✗	✗	✗	✗	✓
	Available methods ¶	-	-	5	-	-	1	13	**25**
	Benchmarked methods	6	9	5	5	8	12	13	**15**
Missingness	MCAR	✓	✗	✗	✓	✓	✓	✓	✓
	MAR	✓	✗	✗	✗	✗	✗	✓	✓
	MNAR	✓	✗	✗	✗	✗	✗	✓	✓
	BO	✗	✗	✗	✗	✗	✗	✗	✓
	Native	✗	✗	✓	✗	✗	✓	✗	✓
Datasets	MIMIC†	✗	III	III	III	✗	III	✗	**III/IV**
	eICU	✗	✗	✗	✗	✗	✗	✗	✓
	HIRID	✗	✗	✗	✗	✗	✗	✗	✓
	AUMCdb	✗	✗	✗	✗	✗	✗	✗	✓
	Other medical	✓	✓	✓	✓	✓	✗	✓	✓
Functionality	Hyperparameter Tuning	✓	✗	✗	✓	✗	✗	✗	✓
	Code Availability	✗	✗	✓	✗	✗	✓	✓	✓
	Time-Series Data	✗	✓	✗	✗	✗	✗	✓	✓
	Data Amputation	✗	✗	✗	✗	✗	✗	✗	✓
	Extensibility	✗	✗	✗	✗	✗	✗	✓	✓

B Extended Results

Note that we aggregate the means and standard deviations over several different runs. We aim to get a comprehensive result summarization with this method.

Figure 2 and Figure 3 display the results in the same manner as Figure 1 in the main text, but for RMSE and JSD.

Table 5 shows the missingness results averaged over each dataset (row) and missingness mechanism (grouped columns) for every metric (individual columns). Table 6 shows the RMSE and MAE for each imputation metric over missingness quantities.

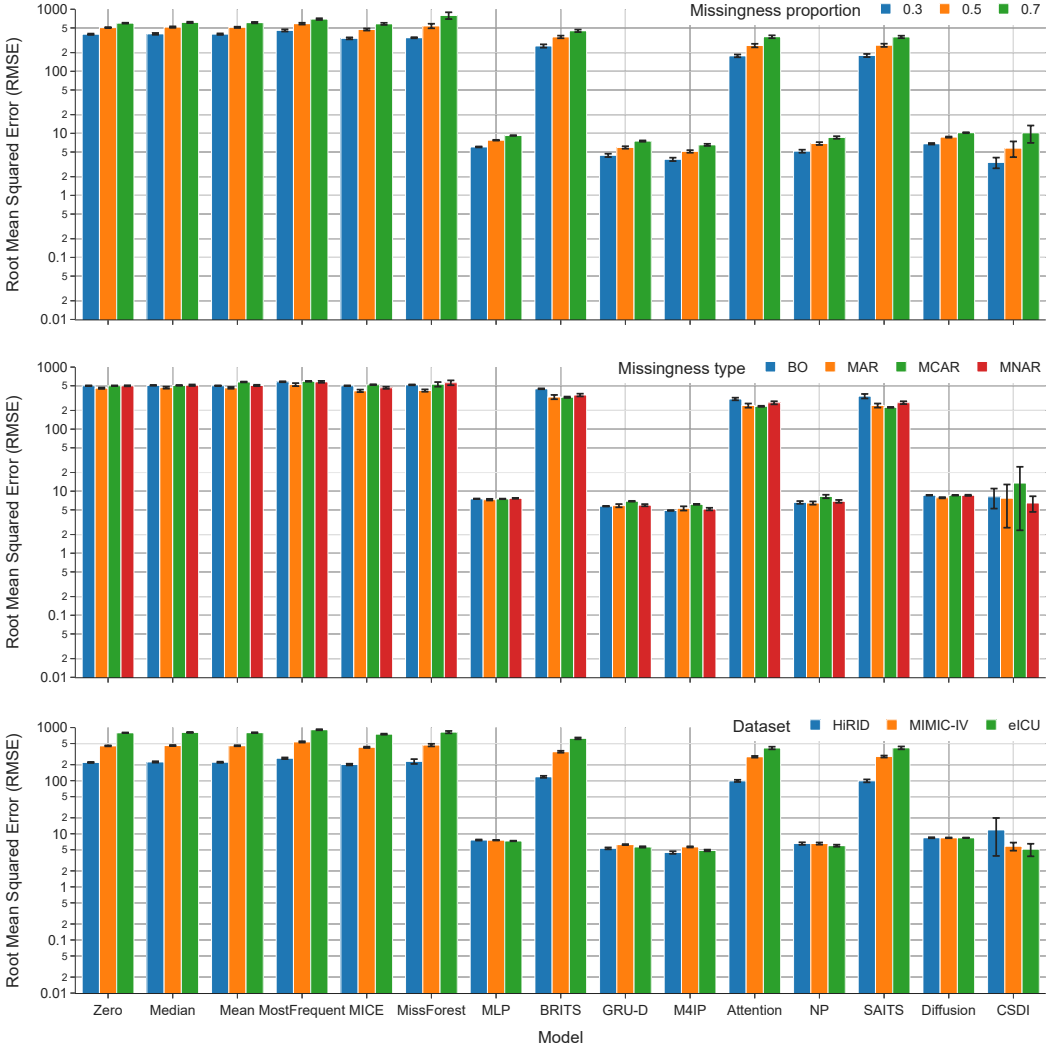


FIGURE 2: *Performance in RMSE across the selected imputation methods in three dimensions.* Note that we use a log scale for readability. **Top:** imputation methods separated by missingness proportion for MNAR. **Middle:** aggregated performance per missingness type. **Bottom:** aggregated performance for each dataset.

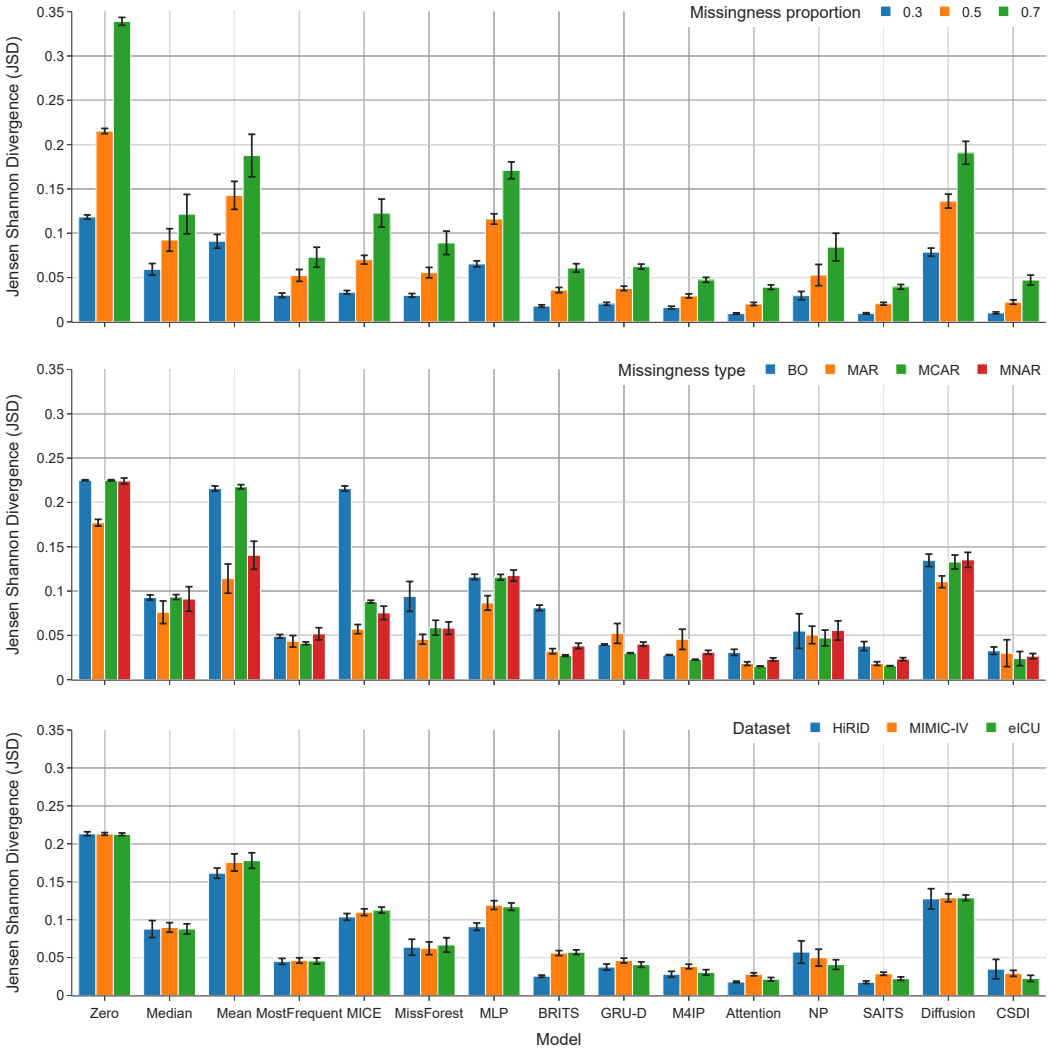


FIGURE 3: *Performance in JSD across the selected imputation methods in three dimensions.* **Top:** imputation methods separated by missingness proportion for MNAR. **Middle:** aggregated performance per missingness type. **Bottom:** aggregated performance for each dataset.

C Data characteristics

We have analyzed the datasets we used in this work to discover patterns that might be interesting for imputation performance.

Figure 4 shows the missingness correlation of the features of each dataset. We can see that some features are heavily missing at the same timestep, depending on the dataset. This can have impact on the result of multiple imputation methods.

Figure 5 explores the concept of informative missingness: we observe a generally higher missingness for survivors than for non-survivors, although this difference is not dramatic. We might ascribe this to a clinical decision to monitor patients in a worse state more frequently.

Figure 6 models a single patient and the imputed values for different models for the six vital signs we impute.

TABLE 5: *Results of all benchmarked imputation methods aggregated by missingness type averaged by dataset.*

Type	MCAR			MAR			MNAR			BO		
	RMSE	MAE	JSD	RMSE	MAE	JSD	RMSE	MAE	JSD	RMSE	MAE	JSD
Dataset												
MIIV	325	0.32	**0.07**	305	0.30	**0.06**	362	0.38	0.08	505	0.52	**0.10**
eICU	604	0.33	**0.07**	527	**0.28**	**0.06**	634	**0.36**	**0.07**	978	0.53	**0.10**
HiRID	**159**	**0.32**	**0.07**	**142**	**0.28**	**0.06**	**171**	**0.36**	**0.07**	**244**	**0.50**	**0.10**

TABLE 6: *Base results averaged over four amputation mechanisms and three datasets, grouped by missingness level (30%, 50%, 70%).* We **embolden** the best model per column and those within a standard deviation (±). RMSE: Root Mean Squared Error (↓, i.e., lower is better), MAE: Mean Absolute Error (↓)

	Missingness Quantity					
	30%		50%		70%	
Model	**RMSE**	MAE	**RMSE**	MAE	**RMSE**	MAE
Zero	386.6±8	0.23±0.00	496.4±8.3	0.37±0.00	586.1±8.7	0.52±0.01
Median	391.6±10.2	0.22±0.01	503.9±11	0.37±0.01	597.6±12.2	0.52±0.01
Mean	387.4±8.6	0.23±0.00	498.8±9.6	0.38±0.01	635.7±12.4	0.53±0.01
MostFrequent	445.7±15.2	0.25±0.01	573±16.8	0.41±0.01	678.6±18.8	0.58±0.02
MICE	337.5±8.5	0.18±0.00	467.7±12.5	0.33±0.01	611.4±15.8	0.49±0.01
MissForest	349.1±5.8	0.19±0.00	497.3±23.1	0.36±0.02	672.7±62.6	0.59±0.06
MLP	5.9±0.1	0.20±0.00	7.6±0.1	0.33±0.01	9.0±0.1	0.47±0.01
BRITS	261.6±13.2	0.13±0.01	368.2±14.2	0.24±0.01	458.3±15.9	0.34±0.01
GRU-D	**3.7±0.2**	0.11±0.00	**5.0±0.2**	0.19±0.01	**7.2±0.2**	0.28±0.01
M4IP	4.3±0.2	0.13±0.00	5.8±0.2	0.23±0.01	8.1±0.2	0.33±0.01
Attention	179.4±7.1	**0.08±0.00**	258.1±11.4	**0.15±0.01**	343±18.6	**0.24±0.01**
NP	4.9±0.2	0.16±0.01	6.6±0.3	0.28±0.02	10.1±0.6	0.41±0.02
SAITS	180.8±8.8	**0.08±0.00**	267.7±17.5	**0.16±0.01**	369.1±22	0.28±0.02
Diffusion	6.6±0.1	0.23±0.00	8.5±0.1	0.38±0.00	10.0±0.1	0.53±0.01
CSDI	4.0±1.6	0.10±0.02	6.3±2.5	0.19±0.03	18.0±13.2	0.37±0.14

	hr	map	sbp	dbp	resp	o2sat
hr	1.00	0.62	0.62	0.62	0.40	0.63
map	0.62	1.00	0.99	0.99	0.36	0.54
sbp	0.62	0.99	1.00	1.00	0.36	0.54
dbp	0.62	0.99	1.00	1.00	0.36	0.54
resp	0.40	0.36	0.36	0.36	1.00	0.34
o2sat	0.63	0.54	0.54	0.54	0.34	1.00

(A) eICU

	hr	map	sbp	dbp	resp	o2sat
hr	1.00	0.76	0.69	0.69	0.89	0.84
map	0.76	1.00	0.89	0.89	0.72	0.70
sbp	0.69	0.89	1.00	1.00	0.65	0.64
dbp	0.69	0.89	1.00	1.00	0.65	0.64
resp	0.89	0.72	0.65	0.65	1.00	0.79
o2sat	0.84	0.70	0.64	0.64	0.79	1.00

(B) MIMIC-IV

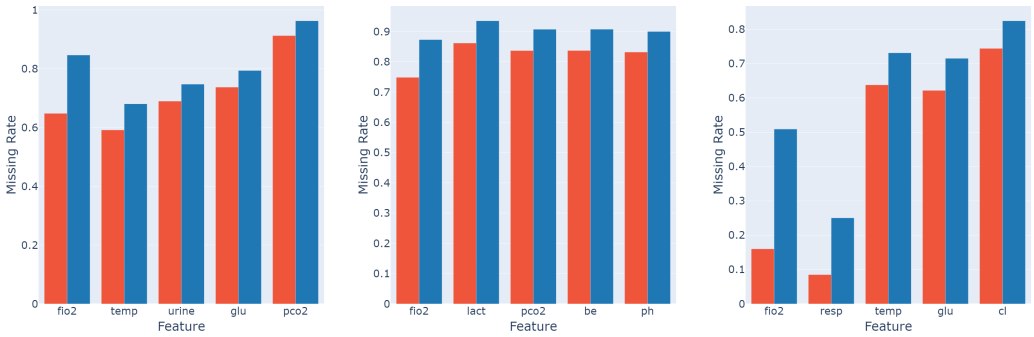
	hr	map	sbp	dbp	resp	o2sat
hr	1.00	0.94	0.65	0.65	0.34	0.90
map	0.94	1.00	0.68	0.68	0.34	0.87
sbp	0.65	0.68	1.00	1.00	0.32	0.60
dbp	0.65	0.68	1.00	1.00	0.32	0.60
resp	0.34	0.34	0.32	0.32	1.00	0.33
o2sat	0.90	0.87	0.60	0.60	0.33	1.00

(C) HiRiD

	hr	map	sbp	dbp	resp	o2sat
hr	1.00	0.64	0.60	0.60	0.18	0.71
map	0.64	1.00	0.94	0.94	0.24	0.54
sbp	0.60	0.94	1.00	1.00	0.24	0.51
dbp	0.60	0.94	1.00	1.00	0.24	0.51
resp	0.18	0.24	0.24	0.24	1.00	0.18
o2sat	0.71	0.54	0.51	0.51	0.18	1.00

(D) HiRiD - 5-minute resolution

FIGURE 4: *Missingness correlation of the selected features for each dataset.*



(A) eICU (B) MIMIC-IV (C) HiRiD

FIGURE 5: *Missingness rate for five features with the highest difference in missing rates between the classes survivor (blue) and non-survivor (red) in each dataset.*

FIGURE 6: *Imputed values across models for a single ICU patient.*

D Extensibility

We have added a range of imputation methods to YAIB [45], including interfaces to existing imputation libraries [8, 18]. Here, we describe the addition of a recently introduced method that uses conditional score-based diffusion models conditioned on observed data: the Conditional Score-based Diffusion Model for Probabilistic Time Series Imputation (CSDI)[44]. To make the process of implementing these models easier, we have created the `ImputationWrapper` class that extends the pre-existing `DLWrapper` (itself a subclass of the `LightningModule` of Pytorch-lightning) with extra functionality.

The CSDI model is a diffusion model that follows the general architecture of conditional diffusion models [14]; It introduces noise into a subset of time series data used as conditional observations to later denoise the data and predict accurate values for the imputation targets. CSDI is based on a U-Net architecture[34] including residual connections.

[44] included two additional features into their model, which are inspired by DiffWave [23]: an attention mechanism and the ability to input side information. The attention mechanism uses transformer layers, as shown in Figure 7. An input with K features, L length, and C channels is reshaped first to apply temporal attention and later reshaped again to apply feature attention. The second additional feature allows side information to be used as input to the model by a categorical feature embedding [44].

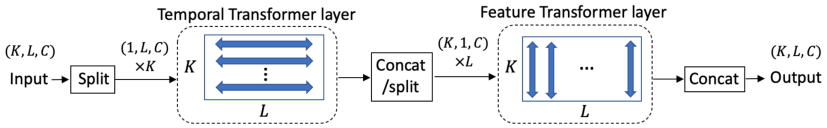


FIGURE 7: The attention mechanism of CSDI adapted from [44].

See Code Listing 1 for the most important implementation code: the model initialization. We note that of this code, very little has been adapted from the original code repository[1] included in the original publication [44].

[1] `https://github.com/ermongroup/CSDI/tree/main`

CODE LISTING 1: *Implementing the CSDI architecture in YAIB*. Note that our implementation is very similar to the original github repository, which demonstrates the flexibility of implementing new models in YAIB.

```
{
    def __init__(
        self, input_size, time_step_embedding_size, feature_embedding_size, unconditional, target_strategy,
              num_diffusion_steps, diffusion_step_embedding_dim, n_attention_heads, num_residual_layers, noise_schedule,
              beta_start, beta_end, n_samples, conv_channels, *args, **kwargs,
    ):
        super().__init__(...)
        self.target_dim = input_size[2]
        self.n_samples = n_samples

        self.emb_time_dim = time_step_embedding_size
        self.emb_feature_dim = feature_embedding_size
        self.is_unconditional = unconditional
        self.target_strategy = target_strategy

        self.emb_total_dim = self.emb_time_dim + self.emb_feature_dim
        if not self.is_unconditional:
            self.emb_total_dim += 1 # for conditional mask
        self.embed_layer = nn.Embedding(num_embeddings=self.target_dim, embedding_dim=self.emb_feature_dim)

        input_dim = 1 if self.is_unconditional else 2
        self.diffmodel = diff_CSDI(
            conv_channels,
            num_diffusion_steps,
            diffusion_step_embedding_dim,
            self.emb_total_dim,
            n_attention_heads,
            num_residual_layers,
            input_dim,
        )

        # parameters for diffusion models
        self.num_steps = num_diffusion_steps
        if noise_schedule == "quad":
            self.beta = np.linspace(beta_start**0.5, beta_end**0.5, self.num_steps) ** 2
        elif noise_schedule == "linear":
            self.beta = np.linspace(beta_start, beta_end, self.num_steps)

        self.alpha_hat = 1 - self.beta
        self.alpha = np.cumprod(self.alpha_hat)
        self.alpha_torch = torch.tensor(self.alpha).float().unsqueeze(1).unsqueeze(1)

}
```