Data Imputation for Multivariate Time-series Data

Conference Paper · October 2023

DOI: 10.1109/KSE59128.2023.10299484

Data Imputation for Multivariate Time-series Data

Lien P. Le[1], Tu T. Do[2], Thu Nguyen[3]
[1] *University of Medicine and Pharmacy at Ho Chi Minh City, Vietnam*
[2] *University of Science, Vietnam National University Ho Chi Minh City, Vietnam*
[3] *Simula Metropolitan, Oslo, Norway*

***Abstract*—Multivariate time-series data are abundant in many application areas, such as finance, transportation, environment, and healthcare. However, for many reasons, missing data points is a common problem, mainly associated with data collected from wearable devices. Missing values negatively impact the performance of data analysis and machine learning algorithms. Various statistical and machine-learning methods have been developed to overcome this challenge, primarily by imputation, i.e., filling in the missing values in the data. In this study, we compare some widely used classical imputation methods such as mean, median imputation, Last Observed Carried Forward (LOCF), K-Nearest Neighbors imputation (KNNI), and some recently developed techniques for time series imputation such as Bidirectional Recurrent Imputation for Time Series (BRITS), Transformer, and Self-attention-based imputation for time series (SAITS). We evaluate these methods on the Crowd-sourced Fitbit dataset on collected activity data through wearables. The results suggest that even though being a classical imputation method, KNNI can be more efficient than some state-of-the-art methods when the missing rate is low to moderate (less than 30%). Meanwhile, at a higher missing rate (greater than or equal to 30%), SAITS is the one that can give the lowest mean absolute error (MAE) with a reasonable execution time.**

***Index Terms*—Multivariate time-series data, Missing values, Imputation model, Wearables, SAITS.**

I. Introduction

In this era of data-driven decision-making, data availability and quality are critical in various fields such as healthcare, finance, meteorology, etc. However, real-world data often suffer from missing values for various reasons, for instance, sensor failures, transmission errors, or gaps in the data collection process [5]. Missing data can severely impact the reliability and accuracy of subsequent analyses and predictive models [13]–[15], especially in the case of time series sensor data.

Missing values are usually categorized into three types: (1) Missing entirely at random (MCAR), where the missing values are independent of any other values; (2) Missing at random (MAR), where the missing values depend only on observed values; (3) Missing not at random (MNAR), where the missing values depend on both observed and unobserved values [7]. This work focuses on the MAR case, which is the most popular type of missingness among the three types above. In this paper, to evaluate all imputation methods, artificial missingness is created by uniformly sampling values to be missing independently. We conducted the comparison under the MCAR assumption. The definition of missing ratio for the datasets is as follows:

$$p = \frac{\text{Number of the missing entries}}{\text{Total number of the entries}}.$$

Imputation, i.e, filling in missing values, has become a crucial step in handling incomplete time series data. A lot of imputation techniques have been developed to estimate missing values, each employing different assumptions, algorithms, and performance metrics. Choosing an appropriate imputation method greatly influences the subsequent analysis and the reliability of the results. Therefore, this paper aims to provide a comprehensive comparative analysis of various missing data imputation techniques for time series sensor data. The primary objective is to evaluate the strengths, limitations, and performance of different imputation methods regarding mean absolute error (MAE) and execution time. In sum, the contribution of this work is as follows:

- We conduct experiments on the Crowd-sourced Fitbit datasets to compare the performance of various time series imputation methods.
- We provide analysis and implications so practitioners can use them as practical guidance.
- We evaluate the performance of the Transformer method and the SAITS one under various numbers of layers.

The remainder of this paper includes four sections as follows: Section II provides an overview of multivariate time-series imputation techniques, along with their strengths and weaknesses. Section III presents the imputation techniques for time series data under comparison in this survey. Section IV outlines the experimental setup and evaluation metrics used to compare these techniques. Section V presents the results and analysis, highlighting the strengths and weaknesses of each method. In the end, Section VI gives the conclusion of the paper with critical findings and directions for future research.

II. Related Works

Many techniques have been proposed to tackle the challenge of missing data imputation in time series sensor data. One widely used imputation technique is the linear interpolation. It assumes a linear relationship between consecutive observed data points and fills in missing values by linearly interpolating between adjacent known values. While simple and intuitive, this method may not be suitable for time-series sensor data with complex patterns or non-linear relationships [11].

Another commonly employed approach is the last observation carried forward (LOCF) method, which propagates the last observed value forward to fill in missing values until a new observation is available. LOCF assumes that the missing data is constant or follows a gradual change. However, this technique may introduce bias and inaccuracies if the missing values are not stationary or the sensor readings exhibit abrupt changes [12].

Some simple imputation methods have attempted to impute missing values statistically using the mean or median of the corresponding feature. The K-nearest neighbor (KNN) imputation algorithm replaces missing values with the average of its neighbors for continuous features and with the majority votes for categorical attributes [16]. Matrix factorization techniques factor the matrix into low-rank matrices and fill the missing values using matrix completion [9]. Multiple Imputation by Chained Equations (MICE) uses multiple imputations by chained equations to fill in the missing values. Some recent works also employed KNN or mean imputation to initialize missing values in multivariate time series; the imputed data were then used to train XGBoost [4] or Ridge regression [10] to impute data in the testing set. However, these methods did not consider the temporal dimension when performing imputation.

Some RNN-based methods have also been proposed to tackle missing value imputation tasks in multivariate time series, such as GRU-D [3], BRITS [2], or RDA [18]. GRU-D [3] utilizes a Gated Recurrent Network (GRU) coupled with a decay mechanism to tackle multivariate missing data in electronic health records (EHR), characterized by two homeostasis properties and decreasing input influence over time. BRITS [2] proposed to handle the missing values in a recurrent dynamical system directly based on the observed data and jointly perform missing value imputation and classification/regression tasks in one neural network. RDA combines RRN with a denoising auto-encoder architecture to impute missing values by considering the missing value as noisy data.

GAN-based networks such as GAIN [21], SeqGAN [22] form a class of Generative Networks that impute missing values by sampling from the underlying distribution of observed data. GAIN [21] adapts the standard GAN architecture. In contrast, GAIN generates the imputed matrix from the masked data matrix, and a random matrix, GAIN's discriminator, tries to distinguish between observed and imputed entries. Since the discriminator is a classifier, we can only derive the gradient when the sequence is entirely generated. SeqGAN [22] overcame this limitation by using the policy gradient to update the generator parameters during intermediate steps.

Additionally, ensemble techniques have been proposed to improve imputation accuracy, such as the research in [19] proposed a method that introduces multiple imputations with an ensemble and compared the proposed method with others that use simple imputation. These methods combine multiple imputation models or techniques to obtain a final imputation. By leveraging the strengths of different approaches, ensemble techniques aim to enhance the robustness and performance of the imputation process.

Recently, research paper [7] proposed the method named SAITS, a self-attention-based method, to impute missing values for multivariate time series. The authors evaluated the algorithm on four public real-world datasets and made comparisons with other methods to conclude that SAITS obtained the best result regarding accuracy and speed training with almost missing rates. However, to our knowledge, no study has been performed to evaluate the performance of SAITS on data from wearable devices such as Fitbit wristbands. With the development of the Internet of Things, these data are more and more popular and give us a lot of information to monitor and improve personal health, but they often have missing values and large gaps for many reasons. Therefore, it is necessary to have an evaluation SAITS and some other methods on multivariate time-series data from wearables to get a comprehensive view and a good guide for practitioners in the healthcare domain.

III. Methods undercomparison

In this section, we detail the methods for missing data imputation under comparison:

1) **Mean Imputation**: This is a simple technique in which each missing value is replaced by the mean of the available ones. With this method, the variability in data is weak. However, the simplicity is its strength. Many researchers have used it, especially when the number of entries is very low or as initialization for some other methods.
2) **Median Imputation**: It is similar to Mean imputation, but the median is utilized instead of the mean.
3) **LOCF (Last Observed Carried Forward)**: This is a naive imputation method that fills missing values with the last observed value of given features in each sample. In case of no previous observation, zero will be utilized.
4) **KNNI (K Nearest Neighbors Imputation)**: The method uses the consideration of the distance between the sample vectors in the dataset's space. For each feature, it utilizes the value of K closest samples that are available in this feature, averages their values in case the feature is continuous, and uses majority voting if the feature is categorical [20].
5) **SAITS (Self-attention-based imputation for time series)**: This is a novel method for the imputation of multivariate time series data. SAITS is constructed based on the self-attention mechanism. SAITS learns missing values from a weighted combination of two diagonally-masked self-attention (DMSA) blocks [7]
6) **BRITS (Bidirectional Recurrent Imputation for Time Series)**: BRITS is a state-of-the-art method for imputing missing values in the case of multiple correlated time series. The technique adapts to bidirectional recurrent neural networks (RNN) without any specific assumption over the data [2].
7) **Transformer**: Transformer is a self-attention-based mechanism for imputing missing values of multivariate time series. It is a joint-optimization training method of imputation and reconstruction for self-attention models [7].

IV. Datasets and experiments

A. Dataset

This study uses the Fitbit Fitness Tracker Dataset [8]. This dataset was generated by respondents who are not younger than 18 years old and regularly wear a Fitbit activity tracker to a distributed survey via Amazon Mechanical Turk between 12.3.2016 and 12.5.2016. The participants connected their personal Fitbit device to an online third-party software system called Fitabase, which allows access to their Fitbit data for the previous month (between 12.3.2016 and 11.4.2016) and the upcoming month (between 12.4.2016 and 12.5.2016) from the date of sign-up [1]. About thirty eligible participants agreed to submit their personal data, including output for manual information, heart rate, physical activity, and sleep monitoring. These variables include daily-level data on calories burned, total steps, total sleep time, distance, and daily active versus sedentary time. We also obtain hourly-level data on calories burned, active versus sedentary time, heart rate, sleep, and step counts. Intraday data's most granular output will include minute-level (step counts) and second-level (heart rate). We have a time series with daily and intraday data. Exporting session ID or timestamp allows individual reports to be parsed to get useful information. Moreover, because of the difference between the types of Fitbit devices the participants used and the individual behaviors, we got the variation in the dataset, including the missing data with different patterns.

B. Experimental Setup

In this paper, we evaluate the data on daily activity. To increase the experiment's reliability, we measure the performance of different methods on two subsets of the dataset. Firstly, we chose the daily activity data of 10 participants from 01.4.2016 to 12.5.2016, which has no missing data. We will call this Dataset 1. This is a small dataset with only 420 samples and 13 features. For Dataset 1, we simulate missing values randomly in each feature with a missing rate ranging from 10% to 80%. To achieve stable results, with every percentage of missing values, we make N simulations, and the average of them was the final outcome.

Dataset 2 contains the daily activity data of all participants in all research time (from 12.3.2016 to 12.5.2016). This dataset has original missing values with a missing rate equal to approximately 36.7%. We evaluate the methods by simulating missing rates from 10% to 80% on observed values. After summing up the original missing values, we have the missing rate total being from 43% to 87%. Once again, with each missing rate, the experiment was performed in N simulations. The final result was obtained by the average of N errors.

In the experiments, we compare the time series imputation methods mentioned in section III together. The implementation of mean, median, and KNN imputation is conducted using the Sklearn package [17]. Meanwhile, BRITS, Transformer, and SAITS are implemented using the Pypots package [6]. Almost all parameters in the methods are default, except the number of layers in the Transformer and SAITS method is chosen 1 instead of 2.

C. Evaluation Metrics

We use MAE (Mean Absolute Error) to evaluate the performance of the imputation methods. The math definition of this metric is presented below.

$$MAE = \frac{\Sigma_{d=1}^{D}\Sigma_{t=1}^{T}|X_t^{d,obs} - X_t^{d,imputed}|}{n}$$

Note that n will be the number of missing values. $X_t^{d,obs}, X_t^{d,imputed}$ is accordingly the original value (the observed value) and the imputed one. T is the number of time steps, and D is the number of dimensions.

V. Results and Inferences

A. MAE

The results of the methods mentioned in Section III for dataset 1 are reported in Tables I and illustrated in Fig. 1. In addition, the results for dataset 2 are reported in Table II and illustrated in Fig. 2, respectively.

Regarding dataset 1 (Fig. 1 and Tables I), which is a small sample dataset with only 420 records and 13 features, we observe that KNNI methods have the best performance with the missing rate under 30%. However, when the missing rate is from 30% to higher, the SAITS model has the lowest mean absolute error. After that, we have the Transformer method for the missing rate from 40%. Especially at an 80% missing rate, BRITS performs approximately as well as SAITS in terms of MAE. Also, for mean and median imputation, the performance is almost the same for all missing rates. With the missing rate under 40%, LOCF has a very high error; it is also the worst method in terms of MAE. However, when the missing rate goes higher to 40%, the error of LOCF begins a light downtrend.

For dataset 2 (Fig. 2), which includes over 2000 records and has a sequence length equaling 62 and some missing gaps, we can observe that the SAITS method outperforms the others. This is consistent with the results on dataset 1 because the actual missing rate (= original missing rate + missing rate for generated missing values) begins from 43%. Similar to dataset 1, mean/ median imputation, BRITS, and LOCF are not as competitive as the others. Mean imputation, median imputation, and LOCF have errors changing lightly every missing rate. BRITS has performance not good in this situation, even though at the missing rate of approximately 80%.

B. Execution Time

The running time of evaluated methods is presented in Fig. 3 and in Fig. 4. The figure shows that mean, median, KNN, and LOCF imputations are fast imputation methods, with less than 0.5 seconds duration under the used missing rate. Transformer, BRITS, and SAITS methods spend more time. But it is still reasonable, with less than 20 seconds at most of the missing rate. Interestingly, KNNI is faster than neural network approaches. This is possible because both datasets are small. Another noteworthy point is that the execution time of SAITS tends to decrease as the missing rate increases slightly.

TABLE I
PERFORMANCE OF IMPUTATION METHODS ON DATASET 1 WITH DIFFERENT MISSING RATES (MEAN ± STANDARD DEVIATION). THE BOLD FONT INDICATES THE BEST PERFORMANCE IN TERMS OF THE MEAN.

Missing rate (%)	Mean	Median	LOCF	KNN	BRITS	Transformer	SAITS
10	0.712 ± 0.008	0.659 ± 0.01	1.183 ± 0.012	**0.269 ± 0.005**	0.799 ± 0.013	0.59 ± 0.011	0.46 ± 0.009
20	0.715 ± 0.005	0.662 ± 0.006	0.883 ± 0.01	**0.302 ± 0.004**	0.602 ± 0.006	0.473 ± 0.009	0.376 ± 0.006
30	0.714 ± 0.004	0.659 ± 0.005	0.774 ± 0.005	0.368 ± 0.005	0.538 ± 0.005	0.449 ± 0.009	**0.357 ± 0.006**
40	0.717 ± 0.002	0.664 ± 0.003	0.731 ± 0.005	0.484 ± 0.004	0.518 ± 0.004	0.451 ± 0.005	**0.371 ± 0.004**
50	0.717 ± 0.002	0.665 ± 0.003	0.7 ± 0.004	0.58 ± 0.005	0.508 ± 0.003	0.47 ± 0.009	**0.39 ± 0.004**
60	0.716 ± 0.002	0.664 ± 0.002	0.682 ± 0.005	0.618 ± 0.003	0.507 ± 0.002	0.493 ± 0.007	**0.422 ± 0.003**
70	0.72 ± 0.002	0.666 ± 0.002	0.671 ± 0.004	0.628 ± 0.003	0.517 ± 0.002	0.534 ± 0.006	**0.468 ± 0.003**
80	0.72 ± 0.002	0.669 ± 0.002	0.665 ± 0.005	0.638 ± 0.003	**0.537 ± 0.003**	0.596 ± 0.012	**0.53 ± 0.005**

TABLE II
PERFORMANCE OF IMPUTATION METHODS ON DATASET 2 WITH ARTIFICIAL MISSING RATES FROM 10% TO 80% (MEAN ± STANDARD DEVIATION). THE BOLD FONT INDICATES THE BEST PERFORMANCE IN TERMS OF THE MEAN.

Artificial missing rate (%)	Missing rate total(%)	Mean	Median	LOCF	KNN	BRITS	Transformer	SAITS
10	43.0	0.695 ± 5.627	0.636 ± 5.145	0.552 ± 4.464	**0.24 ± 1.944**	0.591 ± 4.782	0.315 ± 2.558	**0.247 ± 2.008**
20	49.4	0.695 ± 5.644	0.639 ± 5.189	0.555 ± 4.497	0.316 ± 2.568	0.609 ± 4.951	0.34 ± 2.771	**0.268 ± 2.177**
30	55.7	0.695 ± 5.644	0.636 ± 5.169	0.561 ± 4.555	0.465 ± 3.775	0.611 ± 4.94	0.365 ± 2.98	**0.293 ± 2.373**
40	62.0	0.694 ± 5.637	0.636 ± 5.167	0.565 ± 4.591	0.617 ± 5.007	0.625 ± 5.077	0.396 ± 3.203	**0.311 ± 2.525**
50	68.3	0.695 ± 5.646	0.636 ± 5.174	0.57 ± 4.63	0.69 ± 5.607	0.623 ± 5.074	0.426 ± 3.449	**0.346 ± 2.822**
60	74.7	0.694 ± 5.637	0.636 ± 5.17	0.576 ± 4.673	0.708 ± 5.746	0.636 ± 5.213	0.467 ± 3.798	**0.389 ± 3.175**
70	81.0	0.695 ± 5.641	0.637 ± 5.177	0.586 ± 4.759	0.696 ± 5.651	0.637 ± 5.169	0.527 ± 4.297	**0.443 ± 3.583**
80	87.3	0.696 ± 5.654	0.637 ± 5.168	0.597 ± 4.841	0.674 ± 5.469	0.659 ± 5.346	0.586 ± 4.744	**0.513 ± 4.167**

C. Performance of SAITS and Transformer method under varying parameters settings

The result of evaluating Transformer and SAITS with one layer and two layers for both datasets is presented accordingly in Fig. 5 and in Fig. 6.For both datasets, not only Transformer but also SAITS with one layer has MAE performance better than models with two layers. This is especially emphasized in Transformer. For Dataset 1, both SAITS 1 layer and SAITS 2 layers almost have the same execution time. Meanwhile, the difference between the running time of SAITS with one layer and that of SAITS with two layers is higher for dataset 2. The sample size and the length of the sequence in time series may be the reason for this difference. We also execute the SAITS with different numbers of epochs and observe that the best result is attained at 100 epochs.

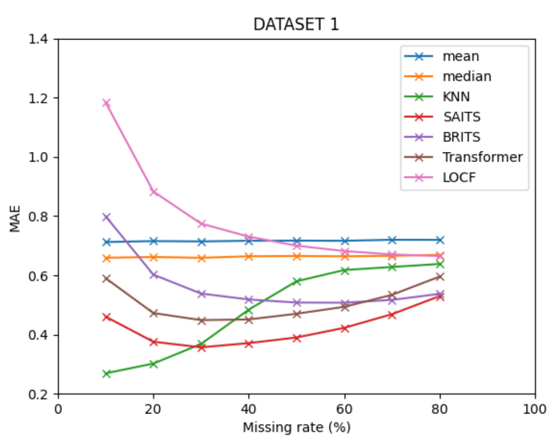

Fig. 1. The performance of various methods on Dataset 1.

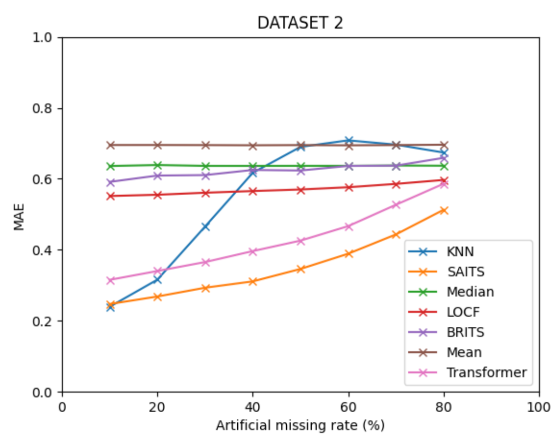

Fig. 2. The performance of various methods on Dataset 2.

VI. CONCLUSIONS

By comparing the performance of Mean Imputation, Median Imputation, KNN Imputation, LOCF method, BRITS model, Transformer, and SAITS on the daily activity data of Fitbit Dataset, we see that KNNI is the best missing data imputation with a missing rate under 30%. However, SAITS gives the lowest error with a reasonable running time when the missing rate increases. SAITS gives a stable result for both datasets. The higher the missing rate value, the better performance SAITS is. The difference in running time is negligible compared to the error efficiency that SAITS provides. Besides that, with a dataset as Fitbit's daily activity, both Transformer and SAITS models perform better with one layer. The methods give smaller MAE by increasing the number of epochs, but there is a limit to this approach.

Based on this research, in the future, we plan to investigate if combining KNNI and SAITS to form an ensembling technique can boost the imputation performance, in terms of MAE and robustness, at both lower and higher missing rates even more than KNNI and SAITS itself. In addition, we plan to compare the performance of imputation techniques for imbalanced missing data and examine more fully other missingness mechanisms.

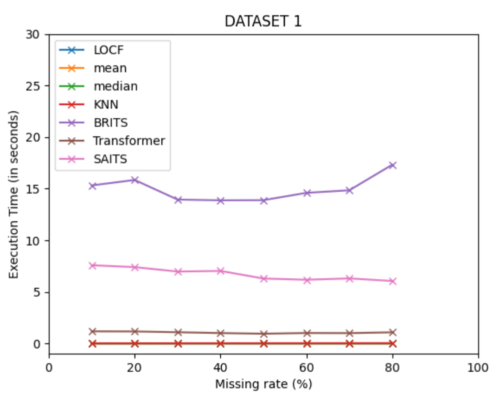

Fig. 3. Running time of various methods on Dataset 1.

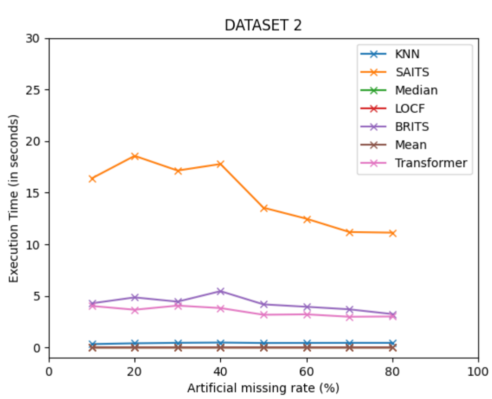

Fig. 4. Running time of various methods on Dataset 2.

REFERENCES

[1] Julia E Brinton, Mike D Keating, Alexa M Ortiz, Kelly R Evenson, and Robert D Furberg. Establishing linkages between distributed survey responses and consumer wearable device datasets: A pilot protocol. *JMIR Res Protoc*, 6(4):e66, Apr 2017.

[2] Wei Cao, Dong Wang, Jian Li, Hao Zhou, Lei Li, and Yitan Li. Brits: Bidirectional recurrent imputation for time series. *Advances in neural information processing systems*, 31, 2018.

[3] Zhengping Che, Sanjay Purushotham, Kyunghyun Cho, David Sontag, and Yan Liu. Recurrent neural networks for multivariate time series with missing values, 2016.

[4] Tianqi Chen and Carlos Guestrin. XGBoost: A scalable tree boosting system. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '16, pages 785–794, New York, NY, USA, 2016. ACM.

[5] Tu T Do, Mai Anh Vu, Hoang Thien Ly, Thu Nguyen, Steven A Hicks, Michael A Riegler, Pål Halvorsen, and Binh T Nguyen. Blockwise principal component analysis for monotone missing data imputation and dimensionality reduction. *arXiv preprint arXiv:2305.06042*, 2023.

[6] Wenjie Du. Pypots: A python toolbox for data mining on partially-observed time series. *arXiv preprint arXiv:2305.18811*, 2023.

[7] Wenjie Du, David Cote, and Yan Liu. SAITS: Self-Attention-based Imputation for Time Series. *Expert Systems with Applications*, 219:119619, 2023.

[8] Robert Furberg, Julia Brinton, Michael Keating, and Alexa Ortiz. Crowd-sourced Fitbit datasets 03.12.2016-05.12.2016, May 2016.

[9] Trevor Hastie, Rahul Mazumder, Jason D Lee, and Reza Zadeh. Matrix completion and low-rank svd via fast alternating least squares. *The Journal of Machine Learning Research*, 16(1):3367–3402, 2015.

[10] A. E. Hoerl and R. W. Kennard. Ridge regression: Biased estimation for nonorthogonal problems. *Technometrics*, 12:55–67, 1970.

[11] Rob J Hyndman and George Athanasopoulos. *Forecasting: principles and practice*. OTexts, 2018.

[12] John M Lachin. Fallacies of last observation carried forward analyses. *Clinical trials*, 13(2):161–168, 2016.

[13] Thu Nguyen, Hoang Thien Ly, Michael Alexander Riegler, and Pål Halvorsen. Principle components analysis based frameworks for efficient missing data imputation algorithms. *arXiv preprint arXiv:2205.15150*, 2022.

[14] Thu Nguyen, Khoi Minh Nguyen-Duy, Duy Ho Minh Nguyen, Binh T Nguyen, and Bruce Alan Wade. Dper: Direct parameter estimation for randomly missing data. *Knowledge-Based Systems*, 240:108082, 2022.

[15] Thu Nguyen, Nhan Thanh Phan, Ha Van Hoang, Pål Halvorsen, Michael A Riegler, and Binh T Nguyen. Pmf: Efficient parameter estimation for data sets with missing data in some features. *Available at SSRN 4260235*.

[16] Thuong HT Nguyen, Bao Le, Phuc Nguyen, Linh GH Tran, Thu Nguyen, and Binh T Nguyen. Principal components analysis based imputation for logistic regression. In *International Conference on Industrial, Engineering and Other Applications of Applied Intelligent Systems*, pages 28–36. Springer, 2023.

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

[18] Esther Rituerto-González and Carmen Peláez-Moreno. End-to-end recurrent denoising autoencoder embeddings for speaker identification. *Neural Computing and Applications*, 33(21):14429–14439, may 2021.

[19] Cao Truong Tran, Mengjie Zhang, Peter Andreae, Bing Xue, and Lam Thu Bui. Multiple imputation and ensemble learning for classification with incomplete data. In George Leu, Hemant Kumar Singh, and Saber Elsayed, editors, *Intelligent and Evolutionary Systems*, pages 401–415, Cham, 2017. Springer International Publishing.

[20] Mai Anh Vu, Thu Nguyen, Tu T Do, Nhan Phan, Pål Halvorsen, Michael A Riegler, and Binh T Nguyen. Conditional expectation for missing data imputation. *arXiv preprint arXiv:2302.00911*, 2023.

[21] Jinsung Yoon, James Jordon, and Mihaela van der Schaar. Gain: Missing data imputation using generative adversarial nets, 2018.

[22] Lantao Yu, Weinan Zhang, Jun Wang, and Yong Yu. Seqgan: Sequence generative adversarial nets with policy gradient, 2017.

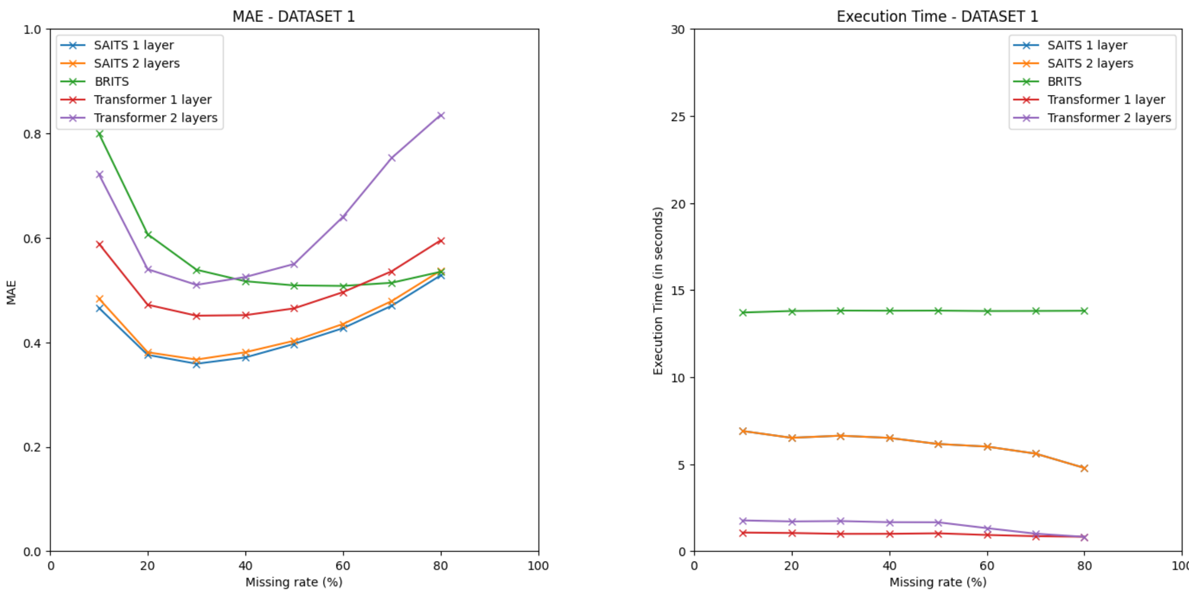

Fig. 5. Evaluating SAITS method with different number of layers on Dataset 1

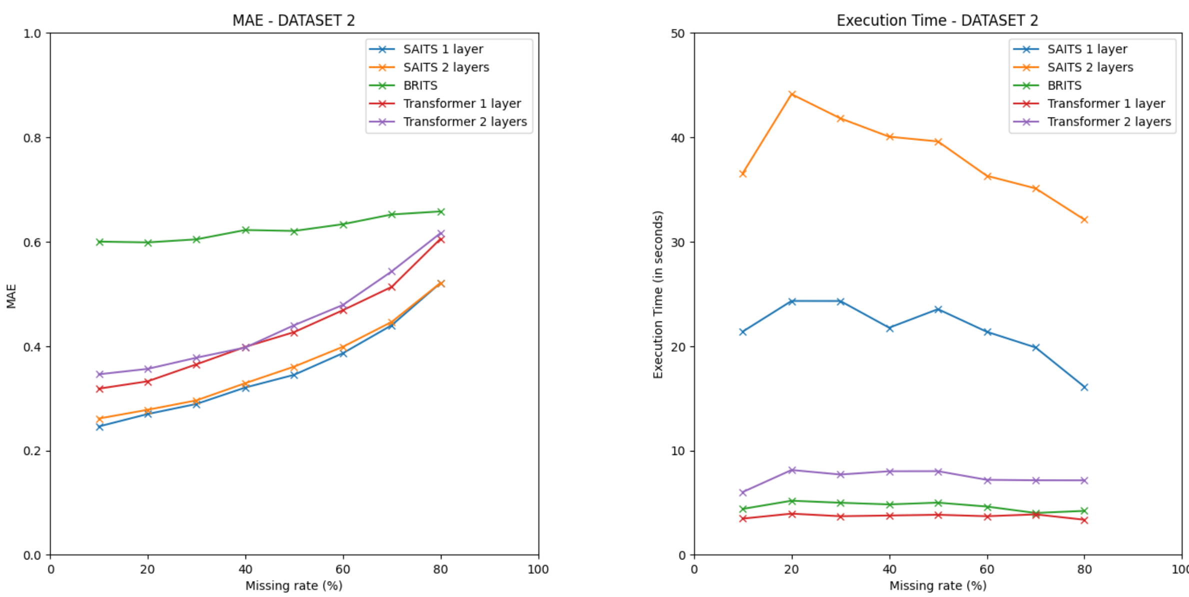

Fig. 6. Evaluating SAITS and Transformer with different number of layers on Dataset 2