Missing Value Imputation for Multi-attribute Sensor Data Streams via Message Propagation (Appendix)

A APPENDIX

Due to the space limit, we show some minute details and relatively less important experimental results/analyses in this appendix.

A.1 The results of MAE on ICU and Airquality.

We complement MAE results of ICU and Airquality datasets in Figure 1.Overall, the results of MAE exhibit similar tendencies as that of MRE in the paper. Referring to (a) and (b), MPIN consistently outperforms the other methods in varying ratios of streams. Besides, the MAE of neural network-based methods would go up when the ratio of streams is small (e.g., 1%). This is due to the lack of training data to impute missing values effectively. In contrast, the effect of the small ratio of streams to FP depends on the dataset. In ICU dataset, the MAE of FP goes up whereas in Airquality dataset, the MAE goes down. We attribute this to the difference in sparsity: the sparsity of ICU dataset is much higher than that of Airquality and thus FP is easier to be affected by the small ratio of concurrent streams in ICU.

Referring to (c) and (d), the MAE of all neural network-based methods have a decreasing tendencies of MAE with the increasing length of a time window. Usually a longer time window means more data instances involved in the time window and thus can contribute more training data for the imputation. Besides, the decreasing tendencies of sequential neural networks such as BRITS and SAITS is sharper than that of MPIN, since sequential models can capture deeper dependencies from longer sequences. However, a longer time window will incur much more time cost for those sequential models, as reported in the paper. Besides, MPIN still achieves the lowest MAE in varying lengths of time windows.

A.2 Effect of Window Length on Wi-Fi Dataset.

As shown in Figures 2, 3 and 4, we complement the experimental results on Wi-Fi dataset regarding the effect of window length. Overall, we can achieve similar conclusions from Wi-Fi dataset as that from the other datasets. First, referring to Figures 2 and 3, MPIN is always the most effective method in varying lengths of time windows. Referring to Figure 4, MPIN is much more efficient than sequential neural network models. Besides, FP is the most efficient method since it simply conducts matrix multiplication for imputation. Compared with FP, MPIN consumes a little more time but achieves much more effectiveness in imputation. Thus, MPIN is the most desirable imputer.

A.3 Effect of Methods of Graph Construction.

We compared two methods for building similarity graphs, namely
R1.W4 the threshold-based and KNN-based methods using the Euclidean distance as the similarity measure. As demonstrated in Figure 5 below, the KNN-based method is better than the threshold-based method in terms of imputation accuracy. This is because the KNN-based method can always ensure each node (i.e., instance) has K edges but the threshold-based method may not. Besides, with the increase of K value, the accuracy of imputation first increases and then drops. In this course, the best performance is achieved when K=10, and thus we set K as 10 in our experiments. When K or the distance threshold is at a low level, meaning that there are few edges for each node in the graph, the accuracy of imputation will be impaired because a node may not be able to capture enough information from neighbors. With the increase of K or threshold value, the accuracy of imputation can be improved. However, when K or threshold is at a too high level, the performance will degrade again. This is because if a node has too many neighboring nodes, some of them may contribute noise since they may be not so similar to the target node.

A.4 Effect of Similarity Function.

Regarding the construction of a similarity graph, we use the Euclidean distance function to calculate the similarity since it can lead to more effective results compared with Cosine similarity function, as shown in Figure 6. In all three datasets, a lower MRE can be obtained by using Euclidean distance as the similarity function.

A.5 Effect of Number of MsgProp Layer.

Referring to Figure 7, we can achieve the best effectiveness result in three datasets when the number of MsgProp layer equals to 2. With only one MsgProp layer, MPIN is hard to capture the correlations among data instances (i.e., nodes) well and cannot utilize the recursive imputation process. Besides, three-layer or even deeper MPIN may encounter the over-smoothing issue [1] due to the use of an internal message-passing module. A two-layer MPIN would be the best option as it can both exploit the correlations among data instances in the similarity graph and avoid the over-smoothing issue caused by deeper structure.

A.6 Effect of Internal Message Passing Module.

We vary the type of internal message passing module from the range
of typical message passing modules such as GAT [6], GCN [5], R3.D2
and GraphSAGE [3] to test their effects on the performance of imputation. The results are shown in Figure 8. Generally, the use of GraphSAGE can achieve a better effect than that of the other two modules. GraphSAGE that follows the principle of sampling-and-aggregation may be better to capture the correlations among data instances to impute their missing values.

A.7 Effect of Value Threshold.

To select a proper value threshold for the data updating strategy, we vary the threshold values from 0.3 to 0.8 and the results of MRE are shown in Figure 9. As we can see, a lower MRE can be achieved

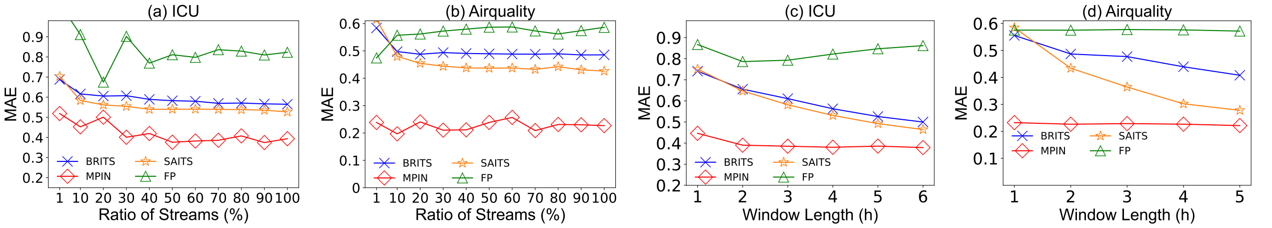

Figure 1: MAE results on ICU and Airquality.

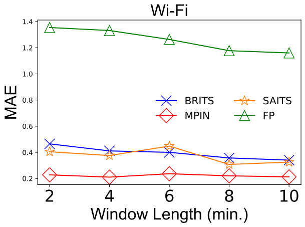

Figure 2: MAE vs. window length.

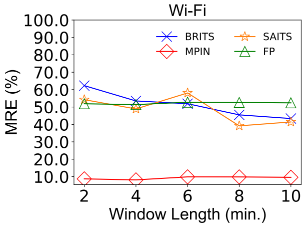

Figure 3: MRE vs. window length.

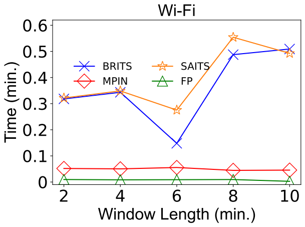

Figure 4: Time vs. window length.

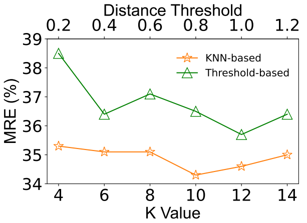

Figure 5: Effect of methods of graph construction.

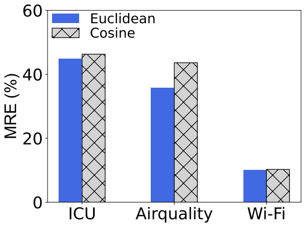

Figure 6: Effect of similarity function.

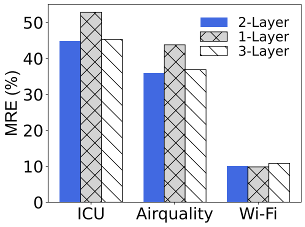

Figure 7: Effect of num of MsgProp layers.

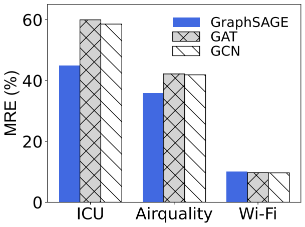

Figure 8: Effect of types of message passing.

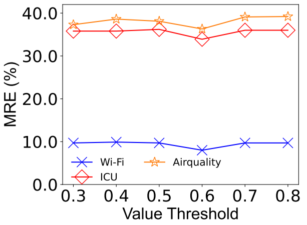

Figure 9: Effect of value threshold.

when the threshold takes a value around 0.6. Besides, the results of MRE appear quite stable in varying thresholds and this gives the flexibility of deciding a proper value threshold.

A.8 Performance on Classification Tasks.

R1.W2 We have conducted a case study on the ICU dataset. Specifically, we trained a classifier on the imputed sensor dataset to predict whether or not a patient in ICU would survive. We use 3 classification metrics (i.e., AUC, F1-score, and accuracy) for comprehensive evaluation. The results are reported in Table 1 below. As we can see, the performance of MPIN on downstream classification tasks is considerably better than the counterpart of the other imputation methods. This further demonstrates the significance of MPIN in imputing missing values in sensor data streams.

Table 1: Performance on Classification Task.

Method	MEAN	KNN	MICE	MF	FP	BRITS	SAITS	MPIN
AUC	0.492	0.525	0.536	0.504	0.749	0.776	0.787	**0.813**
F1-score	0.109	0.121	0.155	0.112	0.353	0.411	0.427	**0.459**
Accuracy	0.750	0.765	0.774	0.784	0.827	0.836	0.832	**0.856**

A.9 Evaluation on Synthetic Datasets.

We have conducted additional experiments on two synthetic datasets that are from an empirical study of imputation [4] (Reference [23] R2.D1 in the paper). The performance of each imputation method on the synthetic datasets is shown in Table 2 below. As we can see, MPIN always achieves much better performance than the other methods in various settings. Only in very few cases, MPIN is the second-best imputer. These experimental results follow similar tendencies as those in the real datasets and further demonstrate the effectiveness of our proposal.

A.10 Proof of Lemmas

This subsection provides proof of Lemma 3 and Lemma 4.

Lemma 3. *$\forall \mathbf{x}_i \in V$, we have $0 \leq \varphi(\mathbf{x}_i) \leq \mathtt{D} - 1$.*

Proof. Referring to Equation 9 in the paper, the importance score $\varphi(\mathbf{x}_i)$ reaches its highest when the first term $OR(\mathbf{x}_i)$ reaches its maximal and the second term $\frac{1}{(|V|-1)} \sum_{\mathbf{x}_k \in V \setminus \mathbf{x}_i} OOR(\mathbf{x}_i, \mathbf{x}_k)$ reaches its minimal; $\mathbf{x}_i$ reaches its lowest when the condition is opposite.

Consider the highest $\varphi(\mathbf{x}_i)$ score. The maximal observation ratio score $OR(\mathbf{x}_i)$ equals $\mathtt{D}$ when $\mathbf{x}_i[d] = 1, \forall 0 \leq d < \mathtt{D}$. As at least one dimension was observed in any of the other instances $\mathbf{x}_k \in V \setminus \mathbf{x}_i$, the minimal value of the second term cannot be lower than 1. Therefore, the highest $\varphi(\mathbf{x}_i)$ is achieved as $\mathtt{D} - 1$.

Table 2: Effectiveness Comparison on Synthetic Datasets (the unit of MRE is %).

Dataset	Synthetic Dataset 1						Synthetic Dataset 2					
Rate	50%		30%		10%		50%		30%		10%	
Metrics	MAE	MRE	MAE	MRE	MAE	MRE	MAE	MRE	MAE	MRE	MAE	MRE
MEAN	0.776	99.42	0.773	99.06	0.77	98.63	0.756	99.62	0.754	99.39	0.752	99.12
KNN	0.686	87.82	0.625	80.15	0.465	59.54	0.609	80.29	0.506	66.69	0.277	36.51
MICE	0.657	83.94	0.557	71.21	0.417	53.34	0.564	74.36	0.435	57.3	0.221	29.13
MF	0.646	82.73	0.555	71.12	0.418	53.61	0.585	77.18	0.477	62.87	0.262	34.47
FP	0.591	75.78	0.465	59.55	0.395	50.62	0.461	60.71	0.31	40.89	0.213	28.04
BRITS	0.478	58.47	0.476	58.33	0.469	57.48	0.219	26.42	0.199	23.98	0.186	22.41
SAITS	0.518	63.42	0.498	61.02	0.487	59.6	0.194	**23.4**	0.191	23.06	0.191	23.04
MPIN	**0.392**	**50.21**	**0.382**	**48.97**	**0.379**	**48.51**	**0.185**	24.44	**0.156**	**20.6**	**0.142**	**18.75**

Consider the lowest $\varphi(\mathbf{x}_i)$ score. The minimal observation ratio score $OR(\mathbf{x}_i)$ equals 1 in the case that only one dimension was observed in $\mathbf{x}_i$. In this case, $\forall \mathbf{x}_k \in V \setminus \mathbf{x}_i$ we have $OOR(\mathbf{x}_i, \mathbf{x}_k) = \mathbf{m}_i^{\mathsf{T}}\mathbf{m}_k \leq 1$ according to Equation 11. Therefore, the lowest $\varphi(\mathbf{x}_i)$ must be no less than $1 - 1 = 0$.

To sum up, we have $0 \leq \varphi(\mathbf{x}_i) \leq \mathtt{D} - 1$. □

Lemma 4. *Given the corresponding gram mask matrix* $\mathbf{M}^{GM}$*, the importance scores of data instances in the set* V *can be computed jointly by*

$$\boldsymbol{\varphi} = \frac{(|V| * \text{diag}^{-1}(\mathbf{M}^{GM}) - \mathbf{M}^{GM} \cdot \mathbf{1}^{|V|\times 1})}{|V| - 1}, \tag{1}$$

where $\boldsymbol{\varphi} \in \mathbb{R}^{|V|\times 1}$ *and* $\boldsymbol{\varphi}[i]$ *captures the importance score of the i-th data instance* $\mathbf{x}_i$ *in the data chunk* $\mathcal{X}_a$, $\text{diag}^{-1}(\cdot)$ *gets the diagonal vector from the input matrix, and* $*$ *refers to the element-wise scalar product.*

Proof. Following literature [2], we have $\mathbf{M}^{\text{GM}}[i,i] = \mathbf{m}_i^{\mathsf{T}}\mathbf{m}_i$ and $\mathbf{M}^{\text{GM}}[i,k] = \mathbf{m}_i^{\mathsf{T}}\mathbf{m}_k$. Let $\boldsymbol{\varphi}'$ be the numerator of Equation 1, we have

$$\begin{aligned}
\boldsymbol{\varphi}' &= |V| * \text{diag}^{-1}(\mathbf{M}^{\text{GM}}) - \mathbf{M}^{\text{GM}} \cdot \mathbf{1}^{|V|\times 1} \\
&=[|V|\mathbf{m}_i^{\mathsf{T}}\mathbf{m}_i - \textstyle\sum_{k=0}^{|V|-1} \mathbf{m}_i^{\mathsf{T}}\mathbf{m}_k : i = 0, \ldots, |V| - 1] \\
&=[(|V| - 1)\mathbf{m}_i^{\mathsf{T}}\mathbf{m}_i - \textstyle\sum_{k=0 \wedge k\neq i}^{|V|-1} \mathbf{m}_i^{\mathsf{T}}\mathbf{m}_k : i = 0, \ldots, |V| - 1] \\
&=[(|V| - 1)OR(\mathbf{x}_i) - \textstyle\sum_{\mathbf{x}_k \in V\setminus \mathbf{x}_i} OOR(\mathbf{x}_i, \mathbf{x}_k) : i = 0, \ldots, |V| - 1].
\end{aligned}$$

Dividing $\boldsymbol{\varphi}'$ by $(|V| - 1)$, we have $\boldsymbol{\varphi} = [\varphi(\mathbf{x}_i) : i = 0, \ldots, |V| - 1]$. Therefore, $\boldsymbol{\varphi}[i]$ equals the importance score of $\mathbf{x}_i$ (see Equation 9). The lemma is thus proved. □

REFERENCES

[1] Deli Chen, Yankai Lin, Wei Li, Peng Li, Jie Zhou, and Xu Sun. 2020. Measuring and relieving the over-smoothing problem for graph neural networks from the topological view. In *Proceedings of the AAAI conference on artificial intelligence*, Vol. 34. 3438–3445.

[2] Petros Drineas, Michael W Mahoney, and Nello Cristianini. 2005. On the Nyström Method for Approximating a Gram Matrix for Improved Kernel-Based Learning. *journal of machine learning research* 6, 12 (2005).

[3] Will Hamilton, Zhitao Ying, and Jure Leskovec. 2017. Inductive representation learning on large graphs. *Advances in neural information processing systems* 30 (2017).

[4] Mourad Khayati, Alberto Lerner, Zakhar Tymchenko, and Philippe Cudré-Mauroux. 2020. Mind the gap: An experimental evaluation of imputation of missing values techniques in time series. In *Proc. VLDB Endow.*, Vol. 13. 768–782.

[5] Thomas N Kipf and Max Welling. 2016. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907* (2016).

[6] Petar Velickovic, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, Yoshua Bengio, et al. 2017. Graph attention networks. *stat* 1050, 20 (2017), 10–48550.