GAIN: Missing Data Imputation using Generative Adversarial Nets

Jinsung Yoon, James Jordon, and Mihaela van der Schaar

Published in ICML (1874 citations to date)

Presented by: Abdullah Mamun

Date: 7 January 2026

Email: a.mamun@asu.edu

GAIN

abdullah-mamun.com

𝕏: @AB9Mamun

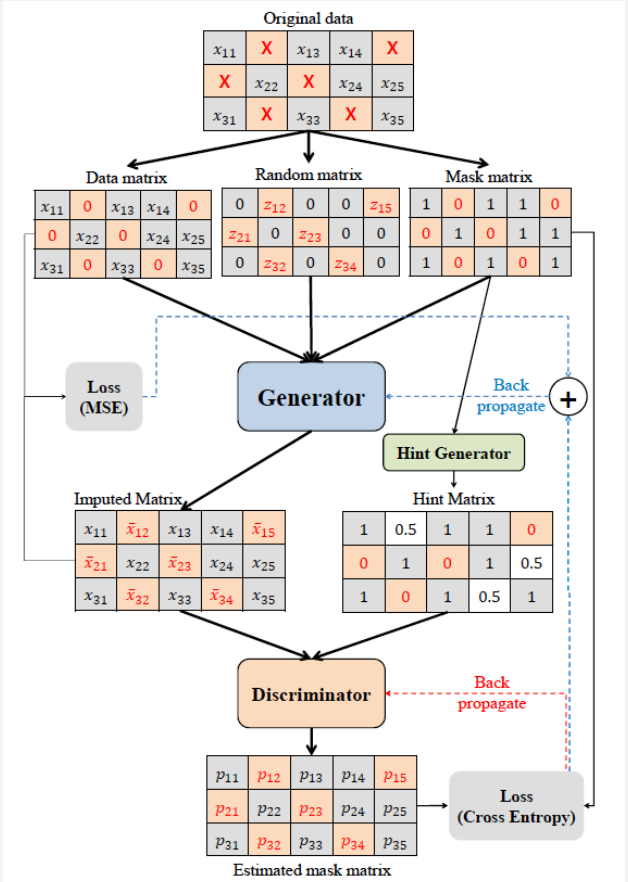


Figure 1. The architecture of GAIN

The Pervasive Problem of Missing Data

Missing data is a universal challenge in almost every field. From medical records where a patient's vital signs weren't recorded, to financial data with gaps in transaction history, incomplete datasets can severely hinder analysis and decision-making.

Why does it matter?

- It compromises the quality of machine learning models.
- It can lead to biased or incorrect conclusions.
- Valuable information is lost, reducing the power of the dataset.

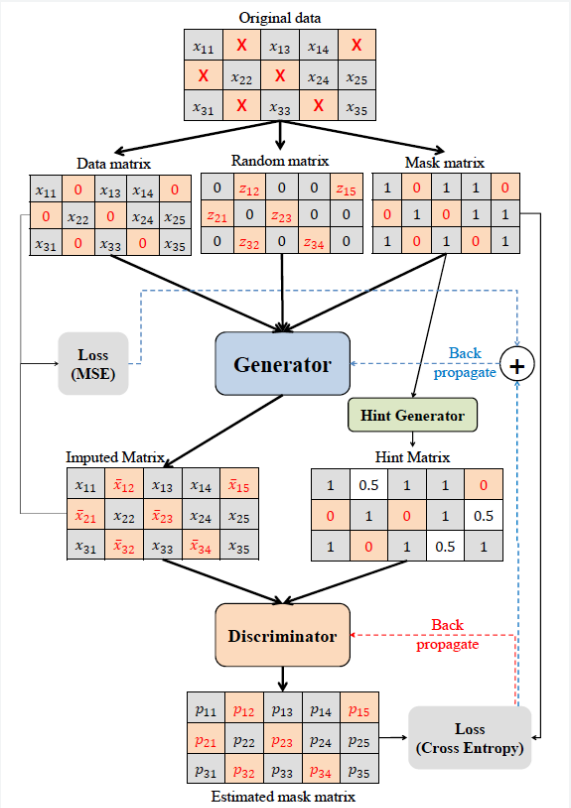


Figure 1. The architecture of GAIN

Understanding the 'Why' Behind Missing Data

Not all missing data is the same. The mechanism causing the data to be missing is crucial.

Missing Completely at Random (MCAR)

The missingness is purely random and doesn't depend on any other data, observed or unobserved. Think of a survey where a few random pages were lost.

Missing at Random (MAR)

The missingness depends on the observed data, but not the missing data itself. For example, men might be less likely to fill out a depression survey, so missingness depends on the 'gender' variable.

Missing Not at Random (MNAR)

The missingness depends on the unobserved data itself. For instance, people with very high incomes might be less likely to disclose their income. This is the hardest case to handle.

GAIN assumes the Missing Completely at Random (MCAR) property.

Introducing GAIN: A New Paradigm

Generative Adversarial Imputation Nets (GAIN) adapts the powerful Generative Adversarial Net (GAN) framework specifically for the task of imputing missing data. It doesn't just fill in the blanks; it learns the underlying data distribution to generate realistic and plausible values.

The Generator (G)

Observes the known data and tries to generate realistic imputations for the missing parts, creating a 'completed' data vector.

The Discriminator (D)

Examines the completed vector and tries to determine which parts were originally observed and which were 'faked' by the generator.

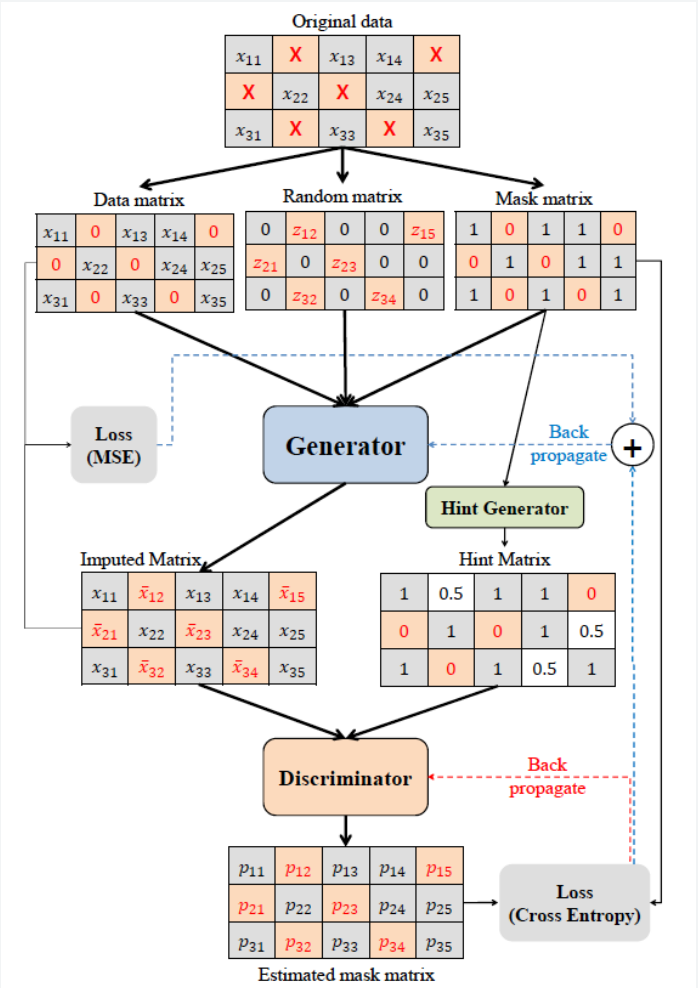


Figure 1. The architecture of GAIN

GAN vs GAIN

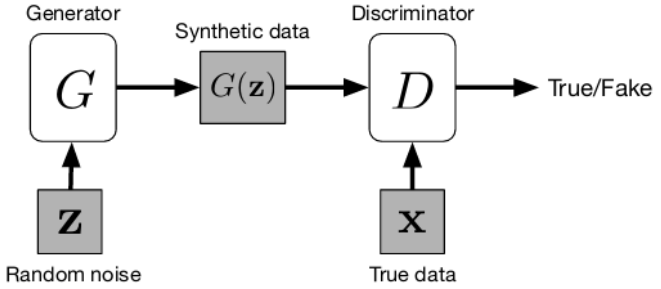


Fig. 1. General structure of a Generative Adversarial Network, where the generator G takes a noise vector $\mathbf{z}$ as input and output a synthetic sample $G(\mathbf{z})$, and the discriminator takes both the synthetic input $G(\mathbf{z})$ and true sample $\mathbf{x}$ as inputs and predict whether they are real or fake.

Huang, H., Yu, P. S., & Wang, C. (2018). An introduction to image synthesis with generative adversarial nets. *arXiv preprint arXiv:1803.04469*

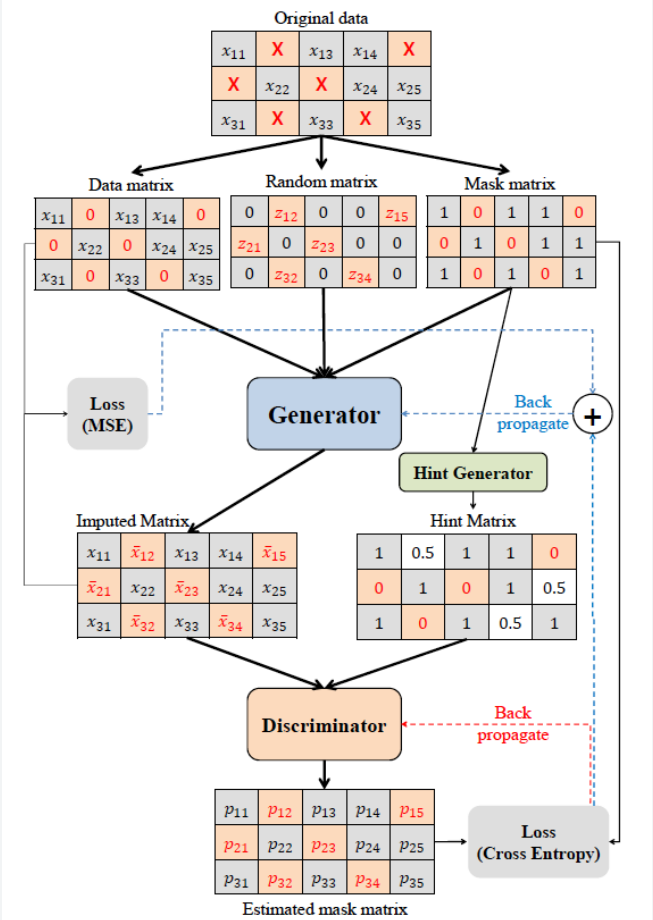


Figure 1. The architecture of GAIN

Yoon, J., Jordon, J., & Schaar, M. (2018, July). Gain: Missing data imputation using generative adversarial nets. In *International conference on machine learning* (pp. 5689-5698). PMLR.

The Hint Mechanism

A vanilla GAN has a limitation. The Generator could learn to create values that are obviously imputed but still globally consistent. The 'hint' is the key innovation to prevent this.

- The hint vector reveals *some* (but not all) of the real mask to the Discriminator.
- This forces the Discriminator to focus on the truly ambiguous, imputed components.
- In turn, this forces the Generator to learn the true data distribution to make its imputations truly believable.
- It guides the adversarial training process towards a meaningful solution.

The discriminator needs to separate the existing values from the masked values from the 0.5 values.

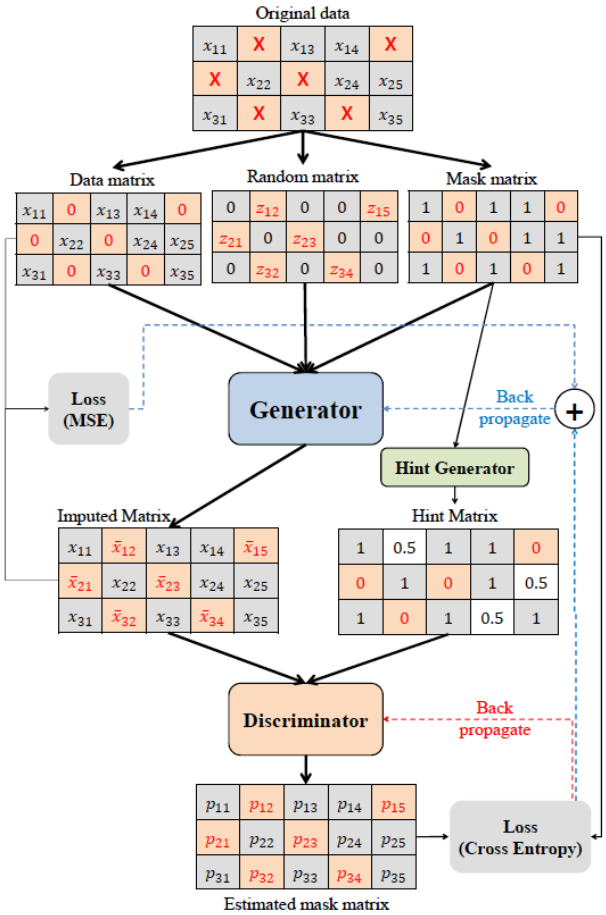


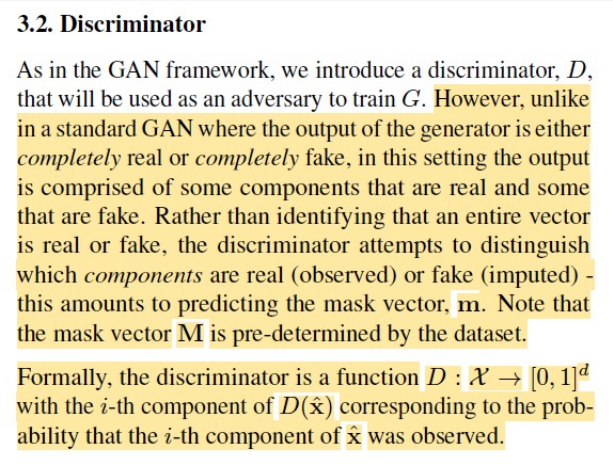
Figure 1. The architecture of GAIN

The GAIN Algorithm: Training Process

The training involves a two-step iterative process, optimizing the Discriminator and Generator in turn.

Step 1: Optimize the Discriminator (D)

With a fixed Generator, D is trained to get better at predicting the mask vector (distinguishing real vs. imputed). This is a standard classification task.



3.2. Discriminator

As in the GAN framework, we introduce a discriminator, D, that will be used as an adversary to train G. However, unlike in a standard GAN where the output of the generator is either *completely* real or *completely* fake, in this setting the output is comprised of some components that are real and some that are fake. Rather than identifying that an entire vector is real or fake, the discriminator attempts to distinguish which *components* are real (observed) or fake (imputed) - this amounts to predicting the mask vector, $\mathbf{m}$. Note that the mask vector $\mathbf{M}$ is pre-determined by the dataset.

Formally, the discriminator is a function $D : \mathcal{X} \rightarrow [0,1]^d$ with the i-th component of $D(\hat{\mathbf{x}})$ corresponding to the probability that the i-th component of $\hat{\mathbf{x}}$ was observed.

Step 2: Optimize the Generator (G)

With a fixed Discriminator, G is trained on a dual-objective: 1) Fool the Discriminator on imputed values (adversarial loss), and 2) Accurately reconstruct the observed values (reconstruction loss).

Experiments

Setup

- Multiple real-world UCI datasets were used (Breast, Spam, Letter, Credit, News).
- Missingness was introduced by randomly removing 20% of data points (MCAR).
- Performance was compared against 5 state-of-the-art imputation methods (MICE, MissForest, Matrix Completion, Auto-encoder, EM).
- Each experiment was run 10 times with 5-fold cross-validation for robustness.

Source of Gains: Why Does GAIN Work?

Ablation studies show that every component of the GAIN architecture contributes to its superior performance. Removing any part degrades the results.

Table 1 is a form of ablation studies. Table 2 is comparison with benchmarks.

GAIN: Missing Data Imputation using Generative Adversarial Nets

Table 1. Source of gains in GAIN algorithm (Mean ± Std of RMSE (Gain (%)))

Algorithm	Breast	Spam	Letter	Credit	News
GAIN	**.0546 ± .0006**	**.0513± .0016**	**.1198± .0005**	**.1858 ± .0010**	**.1441 ± .0007**
GAIN w/o $\mathcal{L}_G$	.0701 ± .0021 **(22.1%)**	.0676 ± .0029 **(24.1%)**	.1344 ± .0012 **(10.9%)**	.2436 ± .0012 **(23.7%)**	.1612 ± .0024 **(10.6%)**
GAIN w/o $\mathcal{L}_M$	.0767 ± .0015 **(28.9%)**	.0672 ± .0036 **(23.7%)**	.1586 ± .0024 **(24.4%)**	.2533 ± .0048 **(26.7%)**	.2522 ± .0042 **(42.9%)**
GAIN w/o Hint	.0639 ± .0018 **(14.6%)**	.0582 ± .0008 **(11.9%)**	.1249 ± .0011 **(4.1%)**	.2173 ± .0052 **(14.5%)**	.1521 ± .0008 **(5.3%)**
GAIN w/o Hint & $\mathcal{L}_M$	.0782 ± .0016 **(30.1%)**	.0700 ± .0064 **(26.7%)**	.1671 ± .0052 **(28.3%)**	.2789 ± .0071 **(33.4%)**	.2527 ± .0052 **(43.0%)**

Conclusion: Full GAIN works better than GAIN with some parts turned off.

Head-to-Head: Imputation Performance (RMSE)

Compared to leading methods, GAIN consistently achieves lower Root Mean Square Error (RMSE), indicating more accurate imputations. Lower is better.

GAIN: Missing Data Imputation using Generative Adversarial Nets

Table 2. Imputation performance in terms of RMSE (Average ± Std of RMSE)

Algorithm	Breast	Spam	Letter	Credit	News
GAIN	**.0546 ± .0006**	**.0513± .0016**	**.1198± .0005**	**.1858 ± .0010**	**.1441 ± .0007**
MICE	.0646 ± .0028	.0699 ± .0010	.1537 ± .0006	.2585 ± .0011	.1763 ± .0007
MissForest	.0608 ± .0013	.0553 ± .0013	.1605 ± .0004	.1976 ± .0015	.1623 ± 0.012
Matrix	.0946 ± .0020	.0542 ± .0006	.1442 ± .0006	.2602 ± .0073	.2282 ± .0005
Auto-encoder	.0697 ± .0018	.0670 ± .0030	.1351 ± .0009	.2388 ± .0005	.1667 ± .0014
EM	.0634 ± .0021	.0712 ± .0012	.1563 ± .0012	.2604 ± .0015	.1912 ± .0011

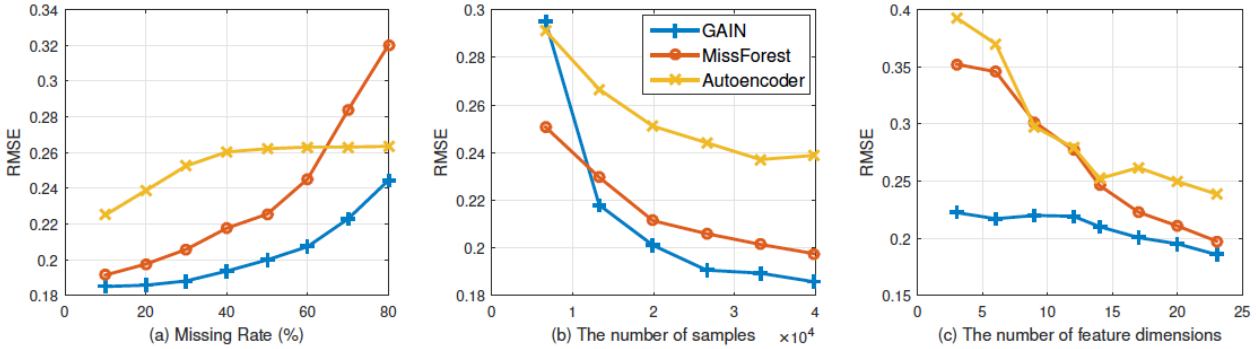


Figure 2. RMSE performance in different settings: (a) Various missing rates, (b) Various number of samples, (c) Various feature dimensions

All the experiments in the Figure 2 are on the Credit dataset.

Prediction Performance

Table 3. Prediction performance comparison

Algorithm	AUROC (Average ± Std)			
	Breast	**Spam**	**Credit**	**News**
GAIN	**.9930 ± .0073**	**.9529 ± .0023**	**.7527 ± .0031**	**.9711 ± .0027**
MICE	.9914 ± .0034	.9495 ± .0031	.7427 ± .0026	.9451 ± .0037
MissForest	.9860 ± .0112	.9520 ± .0061	.7498 ± .0047	.9597 ± .0043
Matrix	.9897 ± .0042	.8639 ± .0055	.7059 ± .0150	.8578 ± .0125
Auto-encoder	.9916 ± .0059	.9403 ± .0051	.7485 ± .0031	.9321 ± .0058
EM	.9899 ± .0147	.9217 ± .0093	.7390 ± .0079	.8987 ± .0157

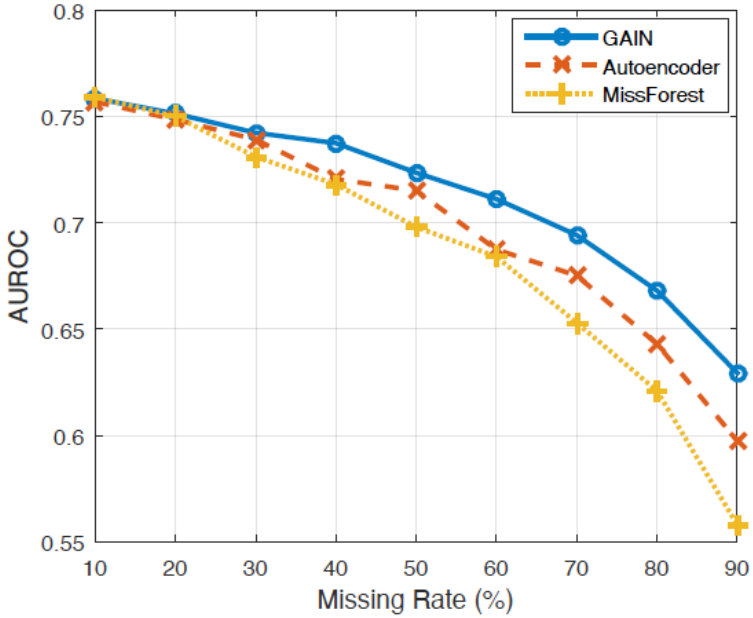


Figure 3. The AUROC performance with various missing rates with Credit dataset

Congeniality: Respecting the Data's Story

A good imputation model should be 'congenial' - it should impute values that respect the original relationships between features and labels. Authors measure this by comparing the parameters of a model trained on the original complete data vs. one trained on imputed data. Lower error is better.

6.5. Congeniality of GAIN

The congeniality of an imputation model is its ability to impute values that respect the feature-label relationship (Meng, 1994; Burgess et al., 2013; Deng et al., 2016). The congeniality of an imputation model can be evaluated by measuring the effects on the feature-label relationships after the imputation. We compare the logistic regression parameters, $\mathbf{w}$, learned from the complete Credit dataset with the parameters, $\hat{\mathbf{w}}$, learned from an incomplete Credit dataset by first imputing and then performing logistic regression.

We report the mean and standard deviation of both the mean bias ($||\mathbf{w} - \hat{\mathbf{w}}||_1$) and the mean square error ($||\mathbf{w} - \hat{\mathbf{w}}||_2$) for each method in Table 4. These quantities being lower indicates that the imputation algorithm better respects the relationship between feature and label. As can be seen in the table, GAIN achieves significantly lower mean bias and mean square error than other state-of-the-art imputation algorithms (from 8.9% to 79.2% performance improvements).

Table 4. Congeniality of imputation models

Algorithm	Mean Bias ($\|\|\mathbf{w} - \hat{\mathbf{w}}\|\|_1$)	MSE ($\|\|\mathbf{w} - \hat{\mathbf{w}}\|\|_2$)
GAIN	**0.3163± 0.0887**	**0.5078± 0.1137**
MICE	0.8315 ± 0.2293	0.9467 ± 0.2083
MissForest	0.6730 ± 0.1937	0.7081 ± 0.1625
Matrix	1.5321 ± 0.0017	1.6660 ± 0.0015
Auto-encoder	0.3500 ± 0.1503	0.5608 ±0.1697
EM	0.8418 ± 0.2675	0.9369 ± 0.2296

GAIN achieves the lowest error, showing it does the best job of preserving the underlying feature-label relationships.

Conclusion & Future Impact

Key Takeaways

- GAIN is a then novel, generative model for missing data imputation that outperforms the state-of-the-art methods of its time.
- Its adversarial architecture, guided by a unique hint mechanism, allows it to learn the true data distribution.
- GAIN's superior imputations lead to better performance in downstream prediction tasks, especially at high missing rates.
- It is highly congenial when compared to the SOTA models of its time, preserving the original structure of the data.

Thank You!

Email: ***a.mamun@asu.edu***

GAIN

abdullah-mamun.com

𝕏: @AB9Mamun