

Diffusion-based Time Series Data Imputation for Microsoft 365

Fangkai Yang*, Wenjie Yin[†], Lu Wang*, Tianci Li*, Pu Zhao*, Bo Liu[§], Paul Wang[§], Bo Qiao*,
Yudong Liu*, Mårten Björkman[†], Saravan Rajmohan[§], Qingwei Lin*, Dongmei Zhang*
Microsoft Research*, Microsoft 365[§], KTH Royal Institute of Technology[†]

ABSTRACT

Reliability is extremely important for large-scale cloud systems like Microsoft 365. Cloud failures such as disk failure, node failure, etc. threaten service reliability, resulting in online service interruptions and economic loss. Existing works focus on predicting cloud failures and proactively taking action before failures happen. However, they suffer from poor data quality like data missing in model training and prediction, which limits the performance. In this paper, we focus on enhancing data quality through data imputation by the proposed Diffusion⁺, a sample-efficient diffusion model, to impute the missing data efficiently based on the observed data. Our experiments and application practice show that our model contributes to improving the performance of the downstream failure prediction task.

CCS CONCEPTS

• Computer systems organization → Cloud computing; • Hardware → Failure prediction.

KEYWORDS

Diffusion model, missing data imputation, cloud failure prediction

1 INTRODUCTION

Microsoft 365 cloud platform is a large-scale online service system and serves millions of customer workloads on a 24/7 basis. It is extremely critical to ensure high reliability as any cloud failure will result in financial loss and degradation of user experience [8, 15, 18]. However, cloud failure, including hardware failure and software failure, is inevitable in large-scale systems [4, 13, 29]. Recent research and works [8, 21, 25, 27] have proposed approaches to predict cloud failures before they actually happen and take actions proactively to mitigate potential failures, thus minimizing the negative impact of cloud failure. Although significant progress has achieved good results in practice, these failure prediction methods still suffer from the issues of data missing [6, 9, 25]. Data missing is a practical and ubiquitous problem in large cloud systems caused by data delay [24], monitoring error [37], etc. In this paper, rather than designing better failure prediction models, we focus on a new perspective of enhancing the data quality by imputing missing data to improve the performance of downstream cloud failure prediction.

There exist a large number of studies of data imputation concerning images and time series data [5, 12, 30, 36]. However, very few focus on time series data imputation in the domain of cloud systems, and rules-based and statistical approaches are commonly used in industry [27]. Most importantly, there lacks an end-to-end evaluation of data imputation with the downstream tasks, and the effect of different data imputation methods is still unexplored connecting to the downstream tasks that utilize the data. In this paper, we leverage the success of the diffusion models [14, 33], which have

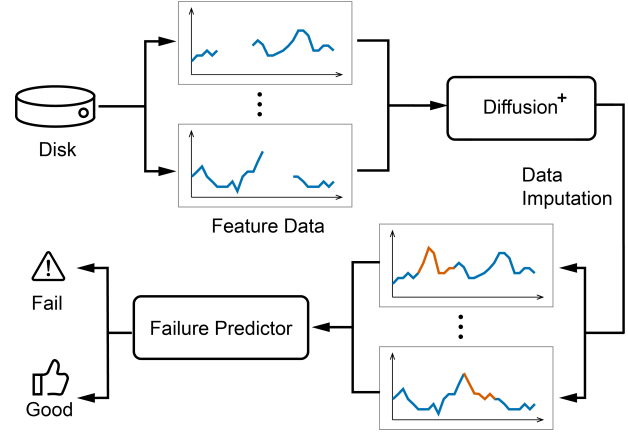


Figure 1: The overview of data imputation with downstream failure prediction tasks.

outperformed state-of-the-art generative models with higher sample quality, and we propose a new diffusion model, *i.e.*, Diffusion⁺, to impute missing data with high efficiency. Figure 1 shows the overview of the process. We use the diffusion model to do data imputation, and the imputed data is fed into the downstream failure prediction task for model training and prediction. We select disk failure prediction as the downstream task since disk failure is one of the most frequent failures in cloud systems [4, 32], and our model can be easily adapted to other downstream tasks in cloud scenarios. Moreover, the slow sampling issue of the diffusion model restricts its application in industry. Inspired by the most recent work [23], we improve the diffusion sampling efficiency with at least 4× speed up without degrading the downstream prediction task.

Our main contributions are summarized as follows:

- We propose a new perspective of improving cloud failure prediction by imputing missing data.
- Inspired by the diffusion model, we propose a new diffusion model with better imputation performance and higher sampling efficiency in the cloud scenario.
- We conduct extensive experiments on industrial data and demonstrate that our model improves the performance on the downstream task.

2 METHODOLOGY

2.1 Problem Formulation

In practice, a disk's status vector is recorded at each timestamp (*e.g.*, hourly), which is a multivariate time series data $x \in \mathbb{R}^{K \times L}$ where K is the number of features and L is the length of time series. At each timestamp l , some data features are missing in x_l , then we can partition x_l into the missing part $x_l^{ms} = \{x_l^k | x_l^k \text{ is missing}\}^{1:K}$ and the observed part $x_l^{ob} = \{x_l^k | x_l^k \text{ is observed}\}^{1:K}$, *i.e.*, $x_l = x_l^{ms} \cup x_l^{ob}$.

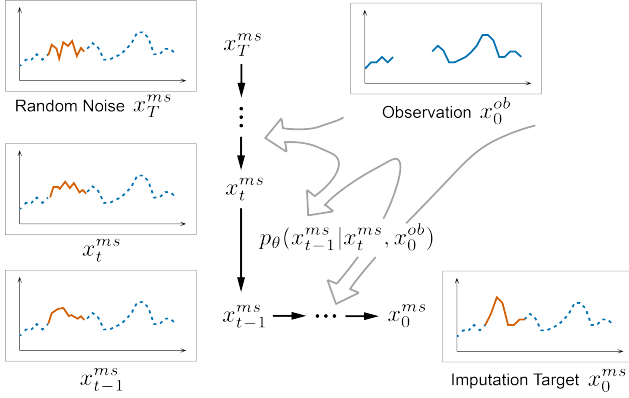


Figure 2: Data imputation with the reverse process of the diffusion model.

Our goal is to do data imputation for $x^{ms} = \{x_l^{ms}\}_{1:L}$ given $x^{ob} = \{x_l^{ob}\}_{1:L}$ for all status feature vectors x , and the imputed status feature vectors are then fed toward downstream prediction tasks, which is trained to predict whether a disk will fail or not.

2.2 Overview of Diffusion⁺ Model

Denosing diffusion probabilistic models (DDPM) [14, 33], known as diffusion models (DM) for brevity, are a class of generative models inspired by non-equilibrium thermodynamics. DMs consist of a forward process and a reverse process. In the forward process, DMs define a fixed Markov chain of T diffusion steps to slowly add noise to the data $x_0 \in \mathbb{R}^{K \times L}$ until the data distribution is close to a standard Gaussian distribution $x_T \in \mathbb{R}^{K \times L}$. Note that the subscripts in x_0 and x_T represent the diffusion step, i.e., $x_0 = \{x_{0,l}\}_{1:L}$, and we omit l for simplicity. The forward process is defined as:

$$q(x_{1:T}|x_0) = \prod_{t=1}^T q(x_t|x_{t-1}), \quad (1)$$

$$q(x_t|x_{t-1}) = \mathcal{N}(x_t; \sqrt{1 - \beta_t}x_{t-1}, \beta_t \mathbf{I}),$$

where $\beta_1, \beta_2, \dots, \beta_T$ are the fixed noise schedulers for controlling the noise scale [14].

On the other hand, in the reverse process, DMs learn to reverse the forward process by denoising to get the desired data distribution from the noise distribution, i.e., sampling from $q(x_{t-1}|x_t)$ will be able to create the true sample x_0 from a Gaussian noise x_T . However, it is non-trivial to estimate $q(x_{t-1}|x_t)$, and we learn to model $p_\theta(\cdot)$ as the approximate estimation. We adopt the conditional diffusion model [36] which uses the observation x_0^{ob} as the condition to generate imputation targets x_0^{ms} . More specifically, the goal of data imputation is to estimate the true conditional data distribution $q(x_0^{ms}|x_0^{ob})$ with a model distribution $p_\theta(x_0^{ms}|x_0^{ob})$, and the missing data x_0^{ms} can be sampled from $p_\theta(\cdot)$ as shown in Figure 2. We model $p_\theta(x_0^{ms}|x_0^{ob})$ with the diffusion model in the reverse process:

$$p_\theta(x_{0:T}^{ms}|x_0^{ob}) = p(x_T^{ms}) \prod_{t=1}^T p_\theta(x_{t-1}^{ms}|x_t^{ms}, x_0^{ob}), \quad x_T^{ms} \sim \mathcal{N}(0, \mathbf{I}), \quad (2)$$

$$p_\theta(x_{t-1}^{ms}|x_t^{ms}, x_0^{ob}) = \mathcal{N}(x_{t-1}^{ms}; \mu_\theta(x_t^{ms}, t|x_0^{ob}), \sigma_\theta(x_t^{ms}, t|x_0^{ob}) \mathbf{I})$$

We define a conditional denoising function ϵ_θ in the reverse process to estimate $\mu_\theta(\cdot)$ and $\sigma_\theta(\cdot)$ of the distribution $p_\theta(x_{t-1}^{ms}|x_t^{ms}, x_0^{ob})$. In particular, $\mu_\theta(x_t^{ms}, t|x_0^{ob}) = \mu^{DDPM}(x_t^{ms}, t, \epsilon_\theta(x_t^{ms}, t|x_0^{ob}))$ and $\sigma_\theta(x_t^{ms}, t|x_0^{ob}) = \sigma^{DDPM}(x_t^{ms}, t)$, where $\mu^{DDPM}(\cdot)$ and $\sigma^{DDPM}(\cdot)$ are the parameterization functions in denoising diffusion probabilistic models (DDPM) [14]. Then, given ϵ_θ and x_0^{ob} , we can sample x_0^{ms} in the reverse process in Equation 2, where ϵ_θ is trainable.

Training. Since we do not have the ground-truth missing values, we first do zero imputation for the missing data, and then we randomly partition the observation x_0^{ob} into two parts: the conditional observation \hat{x}_0^{ob} , and the masked target that needs imputation \hat{x}_0^{ms} . Our model is then trained in a self-supervised learning manner [10] to do data imputation for \hat{x}_0^{ms} given \hat{x}_0^{ob} , and the imputation performance is evaluated on \hat{x}_0^{ms} . With the formulated forward process and the reverse process, the training process optimizes the log-likelihood in the reverse process by maximizing the variational lower bound. The training is performed for all diffusion steps. and it is trained by minimizing the simplified objective function:

$$\min \mathcal{L} = \mathbb{E}_{\epsilon \sim \mathcal{N}(0, \mathbf{I})} \|\epsilon - \epsilon_\theta(\hat{x}_t^{ms}, t|\hat{x}_0^{ob})\|_2^2 \quad (3)$$

Inference. When the training is done, we have good modeling of $p_\theta(\cdot)$. Given the real observation x_0^{ob} as the conditional observation, we could impute the missing data x_0^{ms} with the reverse generation process $x_{t-1}^{ms} \sim p_\theta(x_{t-1}^{ms}|x_t^{ms}, x_0^{ob})$ according to Equation 2.

For each sample with missing data, we generate 100 data imputations and take their median as the final imputed results. The data imputation is conducted over the whole dataset before training the downstream failure prediction models.

Speed up. As shown in Figure 2, DMs suffer from slow sampling as they require a large number of diffusion steps T of running large neural networks to draw one sample [23], which makes it inefficient and impractical for data imputation in industry and becomes a bottleneck for the downstream tasks. Inspired by recent work [23], we speed up the data imputation by reducing the diffusion steps in the reverse process without any further training. The sampling of DMs in the reverse process can be viewed alternatively as solving corresponding ordinary differential equations (ODEs) [16, 34], and the sampling process is done by ODE solvers [3, 23] which results in high-quality and few-step sampling. Specifically, the noise scheduler in $\mu_\theta(\cdot)$ and $\sigma_\theta(\cdot)$ is updated by the ODE solver, and we adopt a uniform step size schedule to determine $M (\ll T)$ steps. Then the diffusion steps T in the reverse process is reduced to M steps.

3 EXPERIMENTS

3.1 Experimental Settings

The data we used for experiments were collected from the Microsoft 365 online service system in recent 6 months. The data is in the SMART format (Self-Monitoring, Analysis and Reporting Technology) [2], which records the disk status and provides important indicators during the lifetime of disks. We predict the disk failure based on 72-hour data. All experiments are performed on a workstation equipped with AMD EPYC 7V12 64-Core CPUs, NVIDIA Tesla T4 GPU with CUDA 10.1, and running Linux (16.04.5) OS.

3.2 Baselines

Following the previous work [12, 27, 36], we use imputation baselines as follows:

Zero imputation (Z): Zero imputation replaces the missing data with zero, which is the most intuitive way.

Forward imputation (F): Forward imputation [19] is a single imputation method that replaces the missing data with the previously observed value.

Linear interpolation (L): Linear interpolation [31] interpolates the missing data by linear curve fitting.

BRITS: BRITS [5] is an RNN-based approach that utilizes a bi-directional recurrent neural network that handles the missing data considering the forward and backward temporal dependency.

Variational Autoencoders (VAE): VAE [12, 17] is a generative model that learns a probability distribution representing the data, and the missing data is sampled from the estimated distribution.

Following the previous work on disk failure prediction [21, 25], we use the downstream prediction baselines: long short-term memory (LSTM) [38], Transformer (Trans) [25], and temporal convolutional neural network (TCNN) [35].

3.3 Experimental Results

In this section, we aim to address three research questions:

- **RQ1: Do the diffusion and Diffusion⁺ models impute missing data effectively?**

As mentioned in Section 2.2, we randomly mask parts of the observation x_0^{ob} as the imputation target \hat{x}_0^{ms} , and we train and evaluate data imputation models with 10%, 50%, 90% masked missing ratio following previous work [36]. Note that Z, F, and L are rule-based methods without training, and we list them for reference.

We first present the quantitative results. We adopt two metrics following previous work [36] to evaluate the performance of data imputation, *i.e.*, MAE (mean absolute error) and CRPS (continuous ranked probability score), where CRPS [28] is usually used to measure the compatibility of an estimated probability distribution with an observation. For the deterministic imputation methods, *i.e.*, Z, F, L, and BRITS, we only use MAE since they are not probabilistic imputation methods. As for probabilistic imputation methods (VAE, Diffusion, and Diffusion⁺), we generate 100 samples for each missing data sample to estimate the probability distribution of the missing data with the metric CRPS. The MAE of the probabilistic imputation methods is computed using the median of 100 generated samples. Note that the data are normalized within each feature dimension in the evaluation. As shown in Table 1, the diffusion model has the lowest MAE, 49%-95% less compared with baselines. It suggests that the diffusion model is more effective in capturing the feature and temporal dependency. The diffusion model also shows the lowest CRPS metric compared with VAE, which indicates its capability of generating more realistic distributions. Diffusion⁺ model has a very close performance as the diffusion model, *i.e.*, second best, in general (excluding models trained under 90% missing ratio). Most imputation approaches have better performance with smaller missing ratios since more observations are available. Thus, we use the models trained under the 10% missing ratio for imputation, *i.e.*, models with the best performance trained with three missing ratios.

Table 1: Data imputation performance evaluated with MAE and CRPS (lower is better). CRPS is only available for probabilistic imputation methods.

Approach	Mssing Ratio (%)					
	10		50		90	
	MAE	CRPS	MAE	CRPS	MAE	CRPS
Z	0.429	—	0.428	—	0.429	—
F	0.047	—	0.057	—	0.111	—
L	0.063	—	0.064	—	0.068	—
BRITS	0.052	—	0.054	—	0.081	—
VAE	0.039	0.613	0.045	0.616	0.075	0.648
Diffusion	0.020	0.049	0.021	0.040	0.053	0.131
Diffusion ⁺	0.025	0.046	0.034	0.068	0.099	0.253

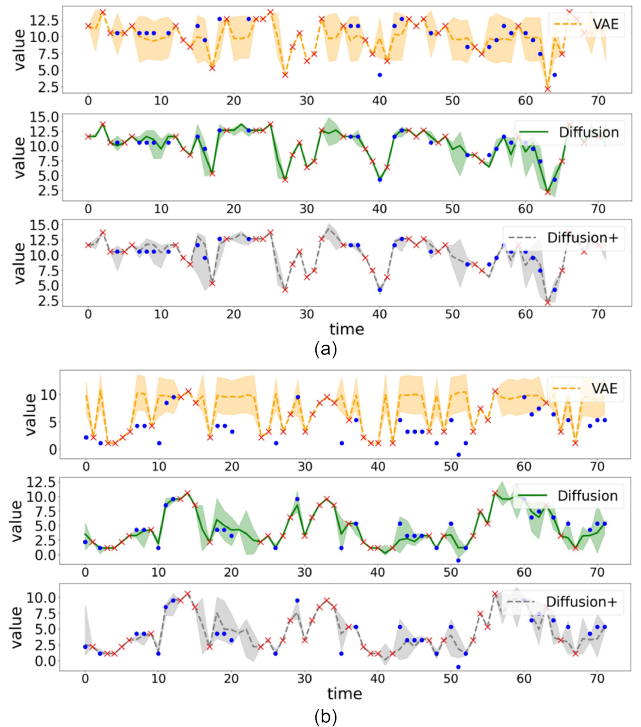


Figure 3: Two data imputation examples of VAE, Diffusion, and Diffusion⁺. Each example is a time series sample of one feature. The red crosses represent observed values and the blue dots represent the masked observation data for imputation targets. The shaded areas are 5% and 95% quantiles and the line is the median value of imputation.

We also provide imputation examples (shown in Figure 3). The diffusion and Diffusion⁺ models generate imputations with high confidence and the imputation distributions tightly cover masked missing targets (blue dots). VAE imputations have larger variations and cannot cover the missing targets.

- **RQ2: Does data imputation contribute to improving the downstream disk failure prediction task?**

We impute the ground-truth missing data in the entire dataset and feed the imputed data to downstream failure prediction tasks. As

Table 2: Failure prediction performance with different data imputation methods on three metrics, *i.e.*, precision, recall, and F1-score.

Approach	Precision	Recall	F1-score
Z+LSTM	60.00	50.45	54.81
F+LSTM	64.69	46.41	54.05
L+LSTM	59.13	44.74	50.94
BRITS+LSTM	61.20	50.22	55.17
VAE+LSTM	62.07	52.84	57.08
Diffusion+LSTM	66.75	55.49	60.60
Diffusion ⁺ +LSTM	65.96	54.23	59.52
Z+Trans	62.84	51.57	56.65
F+Trans	68.15	47.98	56.32
L+Trans	62.87	48.21	54.57
BRITS+Trans	64.81	52.92	58.26
VAE+Trans	66.85	52.45	58.78
Diffusion+Trans	74.05	53.59	62.18
Diffusion ⁺ +Trans	72.01	52.34	60.62
Z+TCNN	60.60	50.00	54.79
F+TCNN	61.05	50.66	55.37
L+TCNN	59.44	47.55	52.83
BRITS+TCNN	60.61	50.32	54.99
VAE+TCNN	60.50	54.64	57.42
Diffusion+TCNN	79.24	48.31	60.03
Diffusion ⁺ +TCNN	72.93	49.78	59.17

shown in Table 2, with all prediction methods, the diffusion imputation model achieves the best performance in precision and F1-score, and also in recall for most cases. In the domain of cloud failure prediction, F1-score is the most important metric [21, 27]. Diffusion⁺ shows a very close performance as the diffusion model in F1-score, and it demonstrates the second best of all the other approaches. Compared with different prediction methods, Trans achieves the best performance in F1-score. Note that we use advanced failure prediction models in practice [21, 25], which have better prediction performance than the prediction baselines.

• **RQ3: Does our Diffusion⁺ model speed up the sampling process in diffusion models?**

Diffusion models suffer from slow sampling issues since generating one sample requires a large number of diffusion steps. Diffusion⁺ model aims to speed up the sampling process with only a few sampling steps without degrading the performance too much. As discussed in RQ1 and RQ2, Diffusion⁺ model achieves similar performance as the diffusion model. Then we conduct the analysis on time cost for imputing each sample. Figure 4 shows the averaged time cost of data imputation for each data sample. As the diffusion step T grows, the time cost for the diffusion model increases accordingly, while Diffusion⁺ has a stable time cost that needs far fewer diffusion steps and it has at least 4× speed up, and the speed up is more obvious with the increase of diffusion steps.

4 APPLICATION IN PRACTICE

We have run our Diffusion⁺ model for one month on Microsoft 365, which contains millions of disks. In particular, our model takes

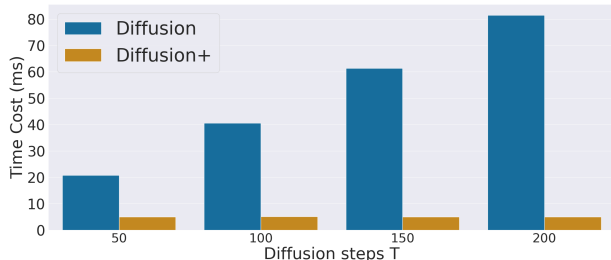


Figure 4: The time cost (ms) for each diffusion imputation.

effect in the data process phase of the current disk failure prediction pipeline [21]. The SMART data is first collected by a data collection service, transferred by a distributed streaming tool, and stored in Azure. Then, our model imputes missing data before sending it to the feature engineering of the downstream prediction tasks. We conduct A/B testing to measure the effectiveness of our model and its contribution to service reliability. We monitor the reduction of virtual machine (VM) interruptions by taking proactive failure mitigation based on the prediction. Compared with the original data process phase without Diffusion⁺, the interruptions have reduced the VM interruptions and enhanced the service reliability to avoid potential financial loss.

5 RELATED WORK

Cloud failure prediction. There exist many studies on cloud failure prediction [4, 13, 29], and they are commonly treated as binary classification problems [21]. They use collected monitoring metrics from services in a time window to predict whether there will be a failure in the near future. They can capture temporal dependency to make a good prediction. However, missing data is a critical issue for these approaches since it requires the prediction models to infer missing information, and it usually results in poor prediction performance [1, 11]. Our paper is orthogonal to these failure prediction methods, and it offers a new perspective to improve cloud failure prediction by enhancing the data quality.

Time series data imputation. Time series data imputation is a rich topic [20]. In particular, deep learning models including RNN-based approaches [5, 7, 22] and generative models [12, 26, 36] can capture the temporal dependency of the time series and generate better data imputation than rule-based and statistical methods. Different from these methods, our paper not only evaluates the imputation quality but also focuses on the end-to-end performance of data imputation with practical industrial problems, *i.e.*, disk failure prediction, and we speed up the diffusion-model-based data imputation to make it applicable in industry.

6 CONCLUSION

In this paper, we focus on enhancing the data quality for disk failure prediction by imputing missing data. We propose our Diffusion⁺ model based on diffusion models which imputes missing data effectively and efficiently. Our experiments on industrial datasets collected in Microsoft 365 and A/B testing show that our model outperforms baselines with fast sampling speed and contributes to enhancing the failure prediction tasks, and then improving the reliability of the Microsoft 365 cloud platform.

REFERENCES

- [1] Juan Miguel Lopez Alcaraz and Nils Strodthoff. 2022. Diffusion-based Time Series Imputation and Forecasting with Structured State Space Models. *arXiv preprint arXiv:2208.09399* (2022).
- [2] Bruce Allen. 2004. Monitoring hard disks with SMART. *Linux Journal* 2004, 117 (2004), 9.
- [3] Kendall Atkinson, Weimin Han, and David E Stewart. 2011. *Numerical solution of ordinary differential equations*. John Wiley & Sons.
- [4] Mirela Madalina Botezatu, Ioana Giurgiu, Jasmina Bogojeska, and Dorothea Wiesmann. 2016. Predicting disk replacement towards reliable data centers. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 39–48.
- [5] Wei Cao, Dong Wang, Jian Li, Hao Zhou, Lei Li, and Yitan Li. 2018. Brits: Bidirectional recurrent imputation for time series. *Advances in neural information processing systems* 31 (2018).
- [6] Chengliang Chai, Lei Cao, Guoliang Li, Jian Li, Yuyu Luo, and Samuel Madden. 2020. Human-in-the-loop outlier detection. In *Proceedings of the 2020 ACM SIGMOD International Conference on Management of Data*. 19–33.
- [7] Zhengping Che, Sanjay Purushotham, Kyunghyun Cho, David Sontag, and Yan Liu. 2018. Recurrent neural networks for multivariate time series with missing values. *Scientific reports* 8, 1 (2018), 1–12.
- [8] Yujun Chen, Xian Yang, Qingwei Lin, Hongyu Zhang, Feng Gao, Zhangwei Xu, Yingnong Dang, Dongmei Zhang, Hang Dong, Yong Xu, et al. 2019. Outage prediction and diagnosis for cloud service systems. In *The World Wide Web Conference*. 2659–2665.
- [9] Supratim Deb, Zihui Ge, Sastry Isukapalli, Sarat Puthenpura, Shobha Venkataraman, He Yan, and Jennifer Yates. 2017. Aesop: Automatic policy learning for predicting and mitigating network service impairments. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 1783–1792.
- [10] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805* (2018).
- [11] Sarah Fletcher Mercaldo and Jeffrey D Blume. 2020. Missing data and prediction: the pattern submodel. *Biostatistics* 21, 2 (2020), 236–252.
- [12] Vincent Fortuin, Dmitry Baranchuk, Gunnar Rätsch, and Stephan Mandt. 2020. Gp-vae: Deep probabilistic time series imputation. In *International conference on artificial intelligence and statistics*. PMLR, 1651–1661.
- [13] Jiechao Gao, Haoyu Wang, and Haiying Shen. 2020. Task failure prediction in cloud data centers using deep learning. *IEEE transactions on services computing* (2020).
- [14] Jonathan Ho, Ajay Jain, and Pieter Abbeel. 2020. Denoising diffusion probabilistic models. *Advances in Neural Information Processing Systems* 33 (2020), 6840–6851.
- [15] Hiranya Jayatilaka, Chandra Krantz, and Rich Wolski. 2017. Performance monitoring and root cause analysis for cloud-hosted web applications. In *Proceedings of the 26th International Conference on World Wide Web*. 469–478.
- [16] Alexia Jolicoeur-Martineau, Ke Li, Rémi Piché-Taillefer, Tal Kachman, and Ioannis Mitliagkas. 2021. Gotta go fast when generating data with score-based models. *arXiv preprint arXiv:2105.14080* (2021).
- [17] Diederik P Kingma and Max Welling. 2013. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114* (2013).
- [18] Sebastien Levy, Randolph Yao, Youjiang Wu, Yingnong Dang, Peng Huang, Zheng Mu, Pu Zhao, Tarun Ramani, Naga Govindaraju, Xukun Li, et al. 2020. Predictive and Adaptive Failure Mitigation to Avert Production Cloud {VM} Interruptions. In *14th USENIX Symposium on Operating Systems Design and Implementation (OSDI 20)*, 1155–1170.
- [19] Roderick JA Little and Donald B Rubin. 2002. Single imputation methods. *Statistical analysis with missing data* (2002), 59–74.
- [20] Todd D Little, Terrence D Jorgensen, Kyle M Lang, and E Whitney G Moore. 2014. On the joys of missing data. *Journal of pediatric psychology* 39, 2 (2014), 151–162.
- [21] Yudong Liu, Hailan Yang, Pu Zhao, Minghua Ma, Chengwu Wen, Hongyu Zhang, Chuan Luo, Qingwei Lin, Chang Yi, Jiaojian Wang, et al. 2022. Multi-task Hierarchical Classification for Disk Failure Prediction in Online Service Systems. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. 3438–3446.
- [22] Yukai Liu, Rose Yu, Stephan Zheng, Eric Zhan, and Yisong Yue. 2019. Naomi: Non-autoregressive multiresolution sequence imputation. *Advances in neural information processing systems* 32 (2019).
- [23] Cheng Lu, Yuhao Zhou, Fan Bao, Jianfei Chen, Chongxuan Li, and Jun Zhu. 2022. DPM-Solver: A Fast ODE Solver for Diffusion Probabilistic Model Sampling in Around 10 Steps. *arXiv preprint arXiv:2206.00927* (2022).
- [24] Sidi Lu, Bing Luo, Tirthak Patel, Yongtao Yao, Devesh Tiwari, and Weisong Shi. 2020. Making Disk Failure Predictions {SMARTer}!. In *18th USENIX Conference on File and Storage Technologies (FAST 20)*. 151–167.
- [25] Chuan Luo, Pu Zhao, Bo Qiao, Youjiang Wu, Hongyu Zhang, Wei Wu, Weihai Lu, Yingnong Dang, Saravanakumar Rajmohan, Qingwei Lin, et al. 2021. NTAM: neighborhood-temporal attention model for disk failure prediction in cloud platforms. In *Proceedings of the Web Conference 2021*. 1181–1191.
- [26] Yonghong Luo, Xiangrui Cai, Ying Zhang, Jun Xu, et al. 2018. Multivariate time series imputation with generative adversarial networks. *Advances in neural information processing systems* 31 (2018).
- [27] Minghua Ma, Yudong Liu, Yuang Tong, Haozhe Li, Pu Zhao, Yong Xu, Hongyu Zhang, Shilin He, Lu Wang, Yingnong Dang, Saravanakumar Rajmohan, and Qingwei Lin. 2022. An Empirical Investigation of Missing Data Handling in Cloud Node Failure Prediction. In *Proceedings of the European Software Engineering Conference and Symposium on the Foundations of Software Engineering (ESEC/FSE)*. 1453 – 1464.
- [28] James E Matheson and Robert L Winkler. 1976. Scoring rules for continuous probability distributions. *Management science* 22, 10 (1976), 1087–1096.
- [29] Justin Meza, Qiang Wu, Sanjev Kumar, and Onur Mutlu. 2015. A large-scale study of flash memory failures in the field. *ACM SIGMETRICS Performance Evaluation Review* 43, 1 (2015), 177–190.
- [30] Irfan Pratama, Adhitya Erna Permanasari, Igi Ardiyanto, and Rini Indrayani. 2016. A review of missing values handling methods on time-series data. In *2016 international conference on information technology systems and innovation (ICITSI)*. IEEE, 1–6.
- [31] ALEXANDER L Read. 1999. Linear interpolation of histograms. *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment* 425, 1-2 (1999), 357–360.
- [32] Sriram Sankar, Mark Shaw, Kushagra Vaid, and Sudhanva Gurumurthi. 2013. Datacenter scale evaluation of the impact of temperature on hard disk drive failures. *ACM Transactions on Storage (TOS)* 9, 2 (2013), 1–24.
- [33] Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. 2015. Deep unsupervised learning using nonequilibrium thermodynamics. In *International Conference on Machine Learning*. PMLR, 2256–2265.
- [34] Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. 2020. Score-based generative modeling through stochastic differential equations. *arXiv preprint arXiv:2011.13456* (2020).
- [35] Xiaoyi Sun, Krishnendu Chakrabarty, Ruirui Huang, Yiquan Chen, Bing Zhao, Hai Cao, Yinhe Han, Xiaoyao Liang, and Li Jiang. 2019. System-level hardware failure prediction using deep learning. In *2019 56th ACM/IEEE design automation conference (DAC)*. IEEE, 1–6.
- [36] Yusuke Tashiro, Jiaming Song, Yang Song, and Stefano Ermon. 2021. CSDI: Conditional score-based diffusion models for probabilistic time series imputation. *Advances in Neural Information Processing Systems* 34 (2021), 24804–24816.
- [37] Jonathan Stuart Ward and Adam Barker. 2014. Observing the clouds: a survey and taxonomy of cloud monitoring. *Journal of Cloud Computing* 3, 1 (2014), 1–30.
- [38] Jianguo Zhang, Ji Wang, Lifang He, Zhao Li, and S Yu Philip. 2018. Layerwise perturbation-based adversarial training for hard drive health degree prediction. In *2018 IEEE International Conference on Data Mining (ICDM)*. IEEE, 1428–1433.