Generative AI for DTN Initialization: High-Fidelity Time-Series Network Data Synthesis via Denoising Diffusion Models

Sangwon Oh
Dept. of Intelligent Electronic and Computer Engineering
Chonnam National University
Gwangju, Korea
osw0788@gmail.com

Juhyun Noh
Dept. of Intelligent Electronic and Computer Engineering
Chonnam National University
Gwangju, Korea
uoahynz@jnu.ac.kr

HoYong Ryu
Network Research Division
Electronics and Telecommunications Research Institute
Daejeon, Korea
hyryu@etri.re.kr

Minsoo Hahn
Dept. of Computational and Data Science
Astana IT University
Astana, Kazakhstan
m.hahn@astanait.edu.kz

Jaehyung Park
Dept. of Intelligent Electronic and Computer Engineering
Chonnam National University
Gwangju, Korea
hyeoung@jnu.ac.kr

Jinsul Kim
Dept. of Intelligent Electronic and Computer Engineering
Chonnam National University
Gwangju, Korea
jsworld@jnu.ac.kr

***Abstract*— AI-driven Digital Twin Networks (DTNs), essential for autonomous network management, face a critical "cold start" problem: initial data scarcity hinders AI model training, leading to unreliable simulations and suboptimal resource allocation. We propose a generative framework using diffusion models to synthesize high-fidelity network metrics (e.g., latency, jitter), specifically designed to capture their complex, non-stationary time-series characteristics. Our diffusion-based method overcomes the training instability and mode collapse issues of traditional GANs, generating data that more accurately preserves the statistical properties of real network traffic. We validate our framework's superiority over GANs through rigorous statistical analysis and performance on downstream tasks like anomaly detection. This work solves the DTN cold-start problem, enabling reliable AI-driven network management from day one and advancing the adoption of resilient, intelligent network virtualization.**

Keywords—Digital Twin Network, Synthetic Data Generation, Generative Adversarial Network, Diffusion Model

I. INTRODUCTION

Driven by advancements in the Internet of Things (IoT), cloud computing, and big data, modern network systems are evolving into complex ecosystems [1]. As network traffic and application demands continue to grow, traditional manual network management has become impractical, compelling a shift towards intelligent, data-driven strategies [2]. In this context, Digital Twin (DT) technology has been integrated with networking to form the Digital Twin Network (DTN) [3, 4]. A DTN establishes bidirectional communication between virtual and physical network spaces, enabling real-time monitoring, optimization, and control [5]. This paradigm allows operators to simulate, analyze, and predict network behavior in a risk-free virtual environment, fundamentally changing how complex networks are managed [6].

The advanced capabilities of DTNs rely on Artificial Intelligence (AI), particularly Machine Learning (ML) and Deep Learning (DL) models [7]. These AI models are essential for processing complex data and predicting future network states to support automated decision-making. However, their performance depends on the availability of large, high-quality datasets for training. This requirement presents a critical bottleneck during the initial deployment of a DTN, a challenge known as the "cold start" problem [8]. The initial scarcity of data severely hampers the training of effective AI models, leading to unreliable simulations, suboptimal resource allocation, and significant delays in deploying intelligent network services.

To mitigate the cold-start problem, network operators often rely on existing data generation methods, primarily network simulators and other generative AI models like Generative Adversarial Networks (GANs). However, each of these approaches has significant limitations :

- **Network Simulators:** While tools such as NS-3 and OMNET++ are widely used, they often struggle with accuracy, speed, and scalability. These computational models may not perfectly capture the real-time dynamics of actual networks and typically lack the real-time, bidirectional communication required for a high-fidelity DTN [6].
- **Generative Adversarial Networks (GANs)**: GANs are used to synthesize network data for privacy or to balance datasets [9]. However, their training is often unstable, and they can suffer from "mode collapse," failing to generate the full diversity of real data [10]. This is a critical flaw for network traffic, as the model may miss rare but important events like anomaly spikes.

To overcome these limitations, this paper introduces a generative model based on Denoising Diffusion Probabilistic Models (DDPMs), a technique adopted in other domains [11]. Diffusion models represent the state-of-the-art in generative AI, demonstrating notable success in producing high-fidelity samples.

We adapt this powerful technology to the unique challenges of network data. Our framework is specifically designed to model the complex characteristics of network metrics, such as latency, jitter, and packet loss, which are non-stationary time-series data exhibiting strong temporal dependencies [12]. We hypothesize that the stable training process and iterative refinement of diffusion models can capture the intricate dynamics of network traffic with higher fidelity than conventional GAN-based approaches.

II. Background and Related Works

A. Fundamentals of Digital Twin Networks

A Digital Twin Network (DTN) is a virtual replica of a physical network, designed for advanced simulation, analysis, and control [4, 5]. As shown Fig 1, DTN operates as an intelligent intermediary between network applications and the physical infrastructure. The architecture consists of three main layers [5] :

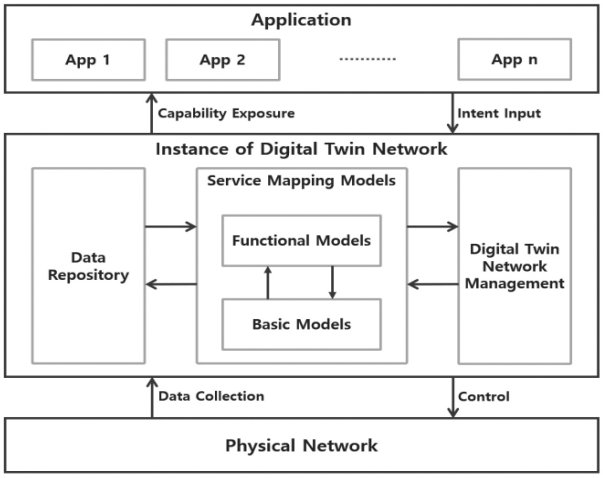

Fig. 1. Digital Twin Network Architecture

- **Physical Network:** This is the foundation, containing the actual network hardware like routers, switches, and servers. It continuously sends real-time operational data to the twin via data collection. In turn, it receives and executes commands from the twin via the control interface.
- **Instance of Digital Twin Network**: This is the core virtual layer that processes data and makes decisions. Its key components include:
  - **(1) Data Repository**: Stores the vast amount of data collected from the physical network, making it available for modeling and analysis.
  - **(2) Service Mapping Models**: This is the "brain" of the DTN. It contains basic models that represent the network's static properties like devices and topology, and functional models that simulate dynamic behaviors like traffic flow and performance.
  - **(3) Digital Twin Network Management**: This component oversees the lifecycle of the twin models and translates application requirements into actionable insights.
- **Application**: This layer contains user-facing services and management tools. Applications provide high-level requirements as Intent Input (e.g., "ensure low latency for video streaming") to the DTN. The DTN then offers its analysis and predictive insights back to the applications through Capability Exposure, enabling smarter, automated network services.

This structure creates a closed-loop system where the DTN continuously learns from the physical network and uses that knowledge to optimize its performance based on application intents.

B. Generative AI for Network Data Synthesis

Generative AI has become a critical tool for addressing data-related challenges in networking, such as data scarcity, privacy, and class imbalance.

- **Generative Adversarial Networks (GANs) in Networking :** A GAN is a class of machine learning frameworks where two neural networks, a generator and a discriminator, compete with each other in a zero-sum game [13]. The generator learns to create plausible data, while the discriminator learns to distinguish the generator's fake data from real data. In the networking domain, GANs have been applied to generate synthetic network and user data to mitigate privacy risks [14], and they have been used to create supplementary training samples to address the non-IID (non-independent and identically distributed) issue in federated learning environments [15]. While effective in certain scenarios, their application is often complicated by the aforementioned challenges of training instability and mode collapse [10].
- **A Primer on Denoising Diffusion Probabilistic Models (DDPMs) :** Diffusion models are a powerful class of generative models that learn to create data by reversing a gradual noising process [11]. This paradigm consists of two processes:
  - **(1) The Forward Process**: A fixed Markov chain that progressively adds Gaussian noise to the input data over a series of timesteps, eventually transforming the data into pure, unstructured noise.
  - **(2) The Reverse Process**: A learned neural network is trained to reverse this process. It learns to iteratively denoise a sample, starting from pure noise, and gradually reconstructs a clean, realistic data sample by predicting the noise that was added at each timestep. This step-by-step refinement process is a key reason for the high sample quality produced by diffusion models and contributes to their more stable training dynamics compared to the adversarial training of GANs.

The task of generating synthetic network data is uniquely challenging. Network metrics such as latency and jitter are not independent data points but are components of a complex, multivariate time-series, making their accurate generation a non-trivial problem [12].

III. Proposed Method and Experimental Design

A. Architecure

The efficacy of diffusion-based generative models is critically dependent on the neural network architecture trained to approximate the reverse denoising process. While conventional DDPMs leverage a U-Net for image generation [11, 16], its architecture, being optimized for 2D spatial data, is ill-suited for the time-series characteristics of the network metrics targeted in this study. Therefore, we adapt and re-engineer the U-Net architecture to effectively model the temporal nature of our data through the following key modifications.

First, we perform a **Dimensionality Transformation**, replacing all 2D operations within the U-Net with their 1D counterparts. Specifically, 2D Convolutional layers are substituted with 1D Convolutional layers that slide along the time axis to capture local temporal patterns. Correspondingly, all downsampling and upsampling operators, such as MaxPooling and Transposed Convolutions, are also adapted to their 1D equivalents.

Second, we implement a mechanism for **Incorporating Conditioning**. The diffusion process must be conditioned on the current timestep t. To achieve this, the scalar value of t is embedded into a high-dimensional vector, which is then integrated into each convolutional block within the U-Net. This design allows the model to perform an optimized denoising operation tailored to the current noise level.

Through these architectural optimizations, our model is tailored to capture the intricate dynamics of network time-series data, enabling the generation of superior-quality synthetic metrics compared to conventional generative models.

B. Experiment Setup

To evaluate our proposed model, we conducted a series of experiments using public benchmark datasets and state-of-the-art baseline models.

- **RCAEval:** The RCAEval dataset is a benchmark for root cause analysis in modern microservice environments [17]. It reflects the complex operational dynamics of cloud-native applications, containing rich telemetry data collected during both normal and fault-injected states. For this study, we extracted key time-series metrics that represent the dynamic state of the system, including CPU Utilization, Memory Usage, and Network Load, to serve as the training and validation data for our generative model.
- **KDDI Cup '99:** This is one of the most well-known benchmark datasets in the field of network intrusion detection, comprising various types of network attacks as well as normal traffic [18]. In this work, the KDDI Cup '99 dataset was not used for direct model training but rather as an evaluation set for a downstream task to assess the practical utility of our synthetic data. Specifically, we validate the quality of the generated data by evaluating how effectively an anomaly detection model, trained on the synthetic data, can classify the attack types present in the KDDI dataset.

All extracted time-series data were preprocessed by partitioning them into fixed-length sequence windows. To ensure stable model training, each metric was subsequently normalized to a [0, 1] range using Min-Max scaling. To demonstrate the superiority of our proposed methodology, we established the following three generative models as baselines for comparison.

- **Vanilla GAN:** The most fundamental form of a GAN, included to establish a performance lower bound [13].
- **TimeGAN (TSGAN):** A state-of-the-art GAN-based model specifically designed for time-series data, combining adversarial and supervised learning to preserve temporal dynamics [19].
- **Proposed DDPM:** The diffusion model optimized for time-series data generation as proposed in this paper, featuring a 1D U-Net architecture conditioned on the diffusion timestep.

IV. Experiment Result

To conduct a multifaceted evaluation of our proposed generative model, we assessed its performance from two critical aspects: Similarity, which measures how well the synthetic data mimics the original data, and Utility, which evaluates its effectiveness as a substitute for real data in an applied task.

To quantitatively assess how accurately the generated data reproduces the statistical distributions of the real data, we defined a Similarity Score based on the Kullback-Leibler (KL) Divergence [20]. The KL-Divergence is a measure of how one probability distribution differs from a second, reference probability distribution; a value of zero indicates that the two distributions are identical. For intuitive comparison, we normalized the KL-Divergence value as formulated in (1), such that a score closer to 1 signifies a smaller divergence and thus a higher degree of statistical similarity.

$$\text{Score}(P, Q) = \frac{1}{1 + \sum_{x \in X} P(x) \log \frac{P(x)}{Q(x)}} \tag{1}$$

Beyond statistical similarity, we evaluated whether the synthetic data can serve as a viable surrogate for real data in a practical DTN operational scenario via a downstream task. For this purpose, we utilized a short-term time-series forecasting model built upon Long Short-Term Memory (LSTM) layers [21]. The evaluation procedure is as follows:

- First, an LSTM forecasting model is trained on the real training dataset and evaluated on the real test dataset to establish a baseline performance, measured in Mean Squared Error (MSE) and Mean Absolute Error (MAE).
- Next, the same trained model is evaluated using the synthetic dataset as input to measure its performance.
- The performance degradation—i.e., the difference in MSE and MAE compared to the baseline—is calculated. A smaller difference is indicative of higher utility. The final Utility Score is defined as shown in (2), (3).

$$\Delta\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i^{\text{real}})^2 - \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i^{\text{syn}})^2 \tag{2}$$

$$\Delta\text{MAE} = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i^{\text{real}}| - \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i^{\text{syn}}| \tag{3}$$

Table 1 summarizes the statistical similarity scores between the synthetic data generated by each model and the original data. The experimental results show that our proposed DDPM model consistently achieves the highest similarity scores across all datasets, significantly outperforming the other baseline models. Table 2 presents the results from the utility evaluation, measured via the downstream time-series forecasting task. The results indicate the degree of performance degradation when using synthetic data, relative

to the baseline performance established with real data. Consequently, the DDPM-generated data led to the least amount of performance degradation in the forecasting model.

TABLE I. RESEULT (SIMILARITY)

Method	Dataset	
	RCAEval	***KDDI Cup '99***
Proposed Diffusion	**0.876 ± 0.012**	**0.913 ± 0.004**
TSGAN	0.751 ± 0.006	0.881 ± 0.008
Vanilla GAN	0.462 ± 0.004	0.824 ± 0.017

TABLE II. RESEULT (UTILITY)

Method	Dataset			
	RCAEval		***KDDI Cup '99***	
	MSE	**MAE**	**MSE**	**MAE**
Proposed Diffusion	**0.110**	**0.265**	**0.032**	**0.143**
TSGAN	0.150	0.309	0.045	0.169
Vanilla GAN	0.270	0.415	0.043	0.165

These findings provide strong evidence that the synthetic data generated by our framework is not only statistically similar to the real data but also holds high practical value, making it a suitable surrogate for training AI models in real-world applications.

V. CONCLUSIONS

This paper addressed the critical "cold start" problem in AI-driven Digital Twin Networks (DTNs), where an initial scarcity of data hinders the training of effective management models. To overcome the limitations of existing network simulators and GAN-based generative models, we introduced a framework based on Denoising Diffusion Probabilistic Models (DDPMs), featuring a U-Net architecture re-engineered for time-series network metric data.

Our comprehensive experiments demonstrated the superiority of the proposed framework. The model significantly outperformed GAN-based baselines in statistical fidelity, and its generated data caused the least performance degradation in a downstream forecasting task, confirming its high practical utility. These findings provide strong evidence that our synthetic data can serve as a viable surrogate for real-world data, effectively solving the initial data scarcity challenge.

While the slower sampling speed of diffusion models remains a limitation, future work will focus on improving sampling efficiency. Furthermore, we plan to extend this research to conditional diffusion-based methods. A promising direction involves using unstructured network log data as a condition to generate corresponding time-series metrics, which would enable high-fidelity simulations of network behavior under specific, log-indicated events.

ACKNOWLEDGMENT

This work was supported by the Institute of Information & communications Technology Planning & Evaluation(IITP) grant funded by the Korea government(MSIT)(Project Name: Development of digital twin-based network failure prevention and operation management automation technology, Project Number: RS-2025-00345030, Contribution Rate, 50%) and IITP(Institute of Information & Coummunications Technology Planning & Evaluation)-ITRC(Information Technology Research Center) grant funded by the Korea government(Ministry of Science and ICT) (IITP-2025-RS-2024-00437718, 50%).

REFERENCES

[1] L. Atzori, A. Iera, and G. Morabito, "The internet of things: A survey," Comput. Netw., vol. 54, no. 15, pp. 2787–2805, Oct. 2010.

[2] Z. Shu, J. Wan, J. Lin, S. Wang, D. Li, S. Rho, and C. Yang, "Traffic engineering in software-defined networking: Measurement and management," IEEE Access, vol. 4, pp. 3246–3256, 2016.

[3] M. Grieves, "Digital twin: Manufacturing excellence through virtual factory replication," White Paper, Florida Institute of Technology, 2014.

[4] R. Poorzare, D. N. Kanellopoulos, V. K. Sharma, P. Dalapati, and O. P. Waldhorst, "Network digital twin toward networking, telecommunications, and traffic engineering: A survey," IEEE Access, vol. 13, pp. 16489–16538, 2025.

[5] Y. Pan, L. Lei, G. Shen, X. Zhang, and P. Cao, "A survey on digital twin networks: Architecture, technologies, applications, and open issues," IEEE Internet Things J., vol. 12, no. 12, pp. 19119–19143, Jun. 2025.

[6] P. Almasan et al., "Network digital twin: Context, enabling technologies, and opportunities," IEEE Commun. Mag., vol. 60, no. 11, pp. 22–27, Nov. 2022.

[7] B. Qin et al., "Machine and deep learning for digital twin networks: A survey," IEEE Internet Things J., vol. 11, no. 21, pp. 34694–34716, Nov. 2024.

[8] A. I. Schein, A. Popescul, L. H. Ungar, and D. M. Pennock, "Methods and metrics for cold-start recommendations," in Proc. 25th Annu. Int. ACM SIGIR Conf. Res. Dev. Inf. Retriev., 2002, pp. 253–260.

[9] M. Ring, S. Wunderlich, D. Grüdl, D. Landes, and A. Hotho, "A survey of network-based intrusion detection data sets," Comput. Secur., vol. 86, pp. 147–167, Oct. 2019.

[10] T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford, and X. Chen, "Improved techniques for training GANs," in Adv. Neural Inf. Process. Syst., vol. 29, 2016.

[11] J. Ho, A. Jain, and P. Abbeel, "Denoising diffusion probabilistic models," in Adv. Neural Inf. Process. Syst., vol. 33, 2020, pp. 6840–6851.

[12] V. Paxson, "End-to-end internet packet dynamics," IEEE/ACM Trans. Netw., vol. 7, no. 3, pp. 277–292, Jun. 1997.

[13] I. Goodfellow et al., "Generative adversarial nets," in Adv. Neural Inf. Process. Syst., vol. 27, 2014.

[14] Y. Lu, X. Huang, K. Zhang, S. Maharjan, and Y. Zhang, "Communication-efficient federated learning and permissioned blockchain for digital twin edge networks," IEEE Internet Things J., vol. 8, no. 4, pp. 2276–2288, Feb. 2021.

[15] X. Zhou et al., "Digital twin enhanced federated reinforcement learning with lightweight knowledge distillation in mobile networks," IEEE J. Sel. Areas Commun., vol. 41, no. 10, pp. 3191–3211, Oct. 2023.

[16] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional networks for biomedical image segmentation," in Proc. Int. Conf. Med. Image Comput. Comput.-Assist. Interv. (MICCAI), 2015, pp. 234–241.

[17] Y. Shen et al., "RCAEval: A benchmark for root cause analysis in microservice systems," 2024, arXiv:2402.17015. [Online]. Available: https://arxiv.org/abs/2402.17015

[18] S. Hettich and S. D. Bay, "The UCI KDD archive," Dept. Inf. Comput. Sci., Univ. California, Irvine, CA, 1999. [Online]. Available: https://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html

[19] J. Yoon, D. Jarrett, and M. Van der Schaar, "Time-series generative adversarial networks," in Adv. Neural Inf. Process. Syst., vol. 32, 2019.

[20] S. Kullback and R. A. Leibler, "On information and sufficiency," Ann. Math. Statist., vol. 22, no. 1, pp. 79–86, Mar. 1951.

[21] S. Hochreiter and J. Schmidhuber, "Long short-term memory," Neural Comput., vol. 9, no. 8, pp. 1735–1780, Nov.