Network Management
Internet-Draft
Intended status: Informational
Expires: 8 January 2026

M. Li
C. Zhou
D. Chen
China Mobile
Q. Wu
Y. Yang
Huawei

7 July 2025

Data Generation and Optimization for Network Digital Twin
draft-li-nmrg-dtn-data-generation-optimization-04

Abstract

Network Digital Twin (NDT) can be used as a secure and cost-effective environment for network operators to evaluate network in various what-if scenarios. Recently, Artificial Intelligence (AI) models, especially neural networks, have been applied for NDT modeling. The quality of deep learning models mainly depends on two aspects: model architecture and data. This memo focuses on how to improve the model quality from the data perspective.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at https://datatracker.ietf.org/drafts/current/.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on 8 January 2026.

Copyright Notice

Table of Contents

1. Introduction

Digital twin is a virtual instance of a physical system (twin) that is continually updated with the physical system's performance, maintenance, and health status data throughout the physical system's life cycle. Network Digital Twin (NDT) is a digital twin that is used in the context of networking [I-D.irtf-nmrg-network-digital-twin-arch]. NDT can be used as a secure and cost-effective environment for network operators to evaluate network in various what-if scenarios. NDT is applicable to various types of networks, such as wireless networks, optical networks, data center networks, Internet of Things (IoT) networks, and vehicular networks.

Artificial Intelligence (AI) models, particularly neural networks (NNs), have proven to be highly effective in modeling complex network environments for various applications, including performance evaluation, traffic prediction, resource allocation, and service self-healing. AI-driven network modeling facilitates the creation of real-time, lightweight, and highly accurate NDT.

The quality of AI models mainly depends on two aspects: model architecture and data. The role of data has recently been highlighted by the emerging concept of data-centric AI [Data-Centric-AI]. This memo focuses on the impact of training data on the model. The quality of training data will directly affect the accuracy and generalization ability of the model. This memo focuses on how to design data generation and optimization methods for NDT modeling, which can generate simulated network data to solve the problem of practical data shortage and select high-quality data from various data sources. Using high-quality data for training can improve the accuracy and generalization ability of the model.

2. Acronyms and Abbreviations

NDT: Network Digital Twin

AI: Artificial Intelligence

AIGC: AI-Generated Content

ToS: Type of Service

OOD: Out-of-Distribution

FIFO: First In First Out

SP: Strict Priority

WFQ: Weighted Fair Queuing

DRR: Deficit Round Robin

BFS: Breadth-First Search

CBR: Constant Bit Rate

3. Requirements

The modeling performance is vital in NDT, which is involved in typical network management scenarios such as planning, construction, operation, optimization, and operation. Recently, some studies have applied AI models to NDT modeling, such as RouteNet [RouteNet], MimicNet [MimicNet] and m3 [m3]. AI is a data-driven technology whose performance heavily depends on data quality.

Data-centric AI [Data-Centric-AI] shifts the focus from model architecture to improving data through various techniques such as data augmentation, self-supervision, data cleaning, data selection, and data privacy. For example, data augmentation can create additional augmented samples. Self-supervised models can be developed without the need for manual labels or features. Data selection methods can help identify the most valuable samples.

In many cases, network data sources are diverse and of varying quality, making it difficult to directly serve as training data for NDT AI models:

* Practical data from production networks: Data from production networks usually have high value, but the quantity, type, and accuracy are limited. Moreover, it is not practical in production networks to collect data under various configurations;

* Network simulators: Network simulators (e.g., NS-3 and OMNeT++) can be used to generate simulated network data, which can solve the problems of quantity, diversity, and accuracy to a certain extent. However, simulation is usually time-consuming. In addition, there are usually differences between simulated data and practical data from production networks, which hinders the application of trained models to production networks;

* Generative AI models: With the development of AI-Generated Content (AIGC) technology, generative AI models (e.g., GPT and LLaMA) can be used to generate simulated network data, which can solve the problems of quantity and diversity to a certain extent. However, the accuracy of the data generated by generative AI models is limited and often has gaps with practical data from production networks.

Therefore, data generation and optimization methods for NDT modeling are needed, which can generate simulated network data to solve the problem of practical data shortage and select high-quality data from multi-source data. High-quality data meets the requirements of high accuracy, diversity, and fitting the actual situation of practical data. Training with high-quality data can improve the accuracy and generalization of NDT performance models.

4. Framework of Data Generation and Optimization

The framework of data generation and optimization for NDT modeling is shown in Figure 1, which includes two stages: the data generation stage and the data optimization stage.

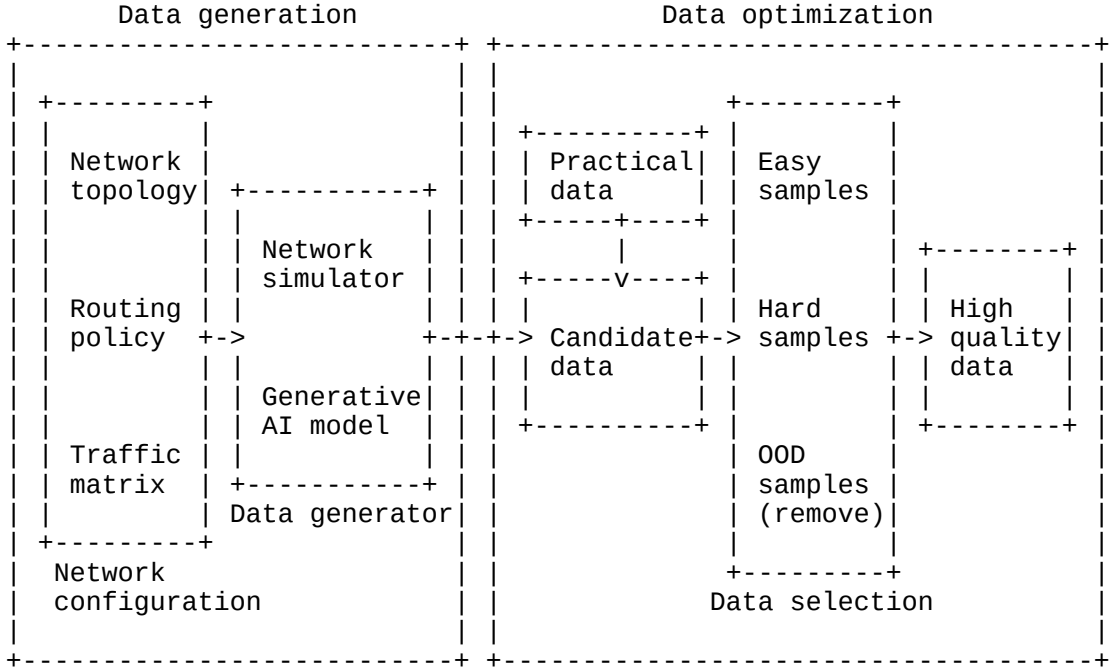

Figure 1: Framework of Data Generation and Optimization for NDT

4.1. Data Generation Stage

The data generation stage aims to generate candidate data (simulated network data) to solve the problem of the shortage of practical data from production networks. This stage first generates network configurations and then imports them into data generators to generate the candidate data.

* Network configurations: Network configurations typically include network topology, routing policy, and traffic matrix. These configurations need to be diverse to cover as many scenarios as possible. Topology configurations include the number and structure of nodes and edges, node buffers' size and scheduling strategy, link capacity, etc. Routing policy determines the path of a packet taking from the source to the destination. The traffic matrix describes the traffic entering/leaving the network, and leaving the footprint in the paths of the network which includes the traffic's source, destination, time and packet size distribution, Type of Service (ToS), etc.

* Data generators: Data generators can be network simulators (e.g., NS-3 and OMNeT++) and/or the generative AI models (e.g., GPT and LLaMA). Network configurations are imported into data generators to generate candidate data.

4.2. Data Optimization Stage

The data optimization stage aims to optimize the candidate data from various sources to select high-quality data.

* Candidate data: Candidate data includes simulated network data generated in the data generation stage and the practical data from production networks.

* Data selection: The data selection module investigates the candidate data to filter out the easy, hard, and Out-of-Distribution (OOD) samples. Hard examples refer to samples that are difficult for the model to accurately predict. During the training process, exposing the model to more hard examples will enable it to perform better on such samples later on. Then the easy samples and hard samples are considered valid samples and added to the training data. OOD samples are considered invalid and removed.

* High-quality data: High-quality data needs to meet the requirements of high accuracy, diversity, and fitting the actual situation of practical data, which can be verified by expert knowledge (such as the ranges of delay, queue utilization, link utilization, and average port occupancy).

5. Data Generation

This section will describe how to generate network configurations, including network topology, routing policy, and traffic matrix. Then these configurations will be imported into data generators to generate the candidate data.

5.1. Network Topology

Network topologies are generated using the Power-Law Out-Degree algorithm, where parameters are set according to real-world topologies in the Internet Topology Zoo.

When the flow rate exceeds the link bandwidth or the bandwidth set for the flow, the packet is temporarily stored in the node buffer. A larger node buffer size means a larger delay and possibly a lower packet loss rate. The node scheduling policy determines the time and order of packet transmission, which is randomly selected from the policies such as First In First Out (FIFO), Strict Priority (SP), Weighted Fair Queuing (WFQ), and Deficit Round Robin (DRR).

A larger link capacity means a smaller delay and less congestion. To cover diverse link loads to get good coverage of possible scenarios, we set the link capacity to be proportional to the total average bandwidth of the flows passing through the link.

5.2. Routing Policy

Routing policy plays a crucial role in routing protocols, which determines the path of a packet from the source to the destination.

* Default: We set the weight of all links in the topology to be the same, that is, equal to 1. Then we use the Dijkstra algorithm to generate the shortest path configuration. Dijkstra algorithm uses Breadth-First Search (BFS) to find the single source shortest path in a weighted digraph.

* Variants: We randomly select some links (the same link can be chosen more than once) and add a small weight to them. Then we use the Dijkstra algorithm to generate a series of variants of the default shortest path configuration based on the weighted graph. These variants can add some randomness to the routing configuration to cover longer paths and larger delays.

5.3. Traffic Matrix

The traffic matrix is very important for network modeling. The traffic matrix can be seen as a network map, which describes the traffic entering/leaving the network, including the source, destination, distribution of the traffic, etc.

We generate traffic matrix configurations with variable traffic intensity to cover low to high loads.

The parameters packet sizes, packet size probabilities, and ToS are generated according to the validation dataset analysis to have similar distributions.

The arrival of packets for each source-destination pair is modeled using one of the time distributions such as Poisson, Constant Bit Rate (CBR), and ON-OFF.

6. Data Optimization

This section will describe how to optimize the data from various sources to filter out high-quality data, which includes the seed sample selection phase and incremental optimization phase.

Candidate data includes simulated network data generated in the data generation stage and real data from production networks. Data optimization supports a variety of selection strategies, including high fidelity, high coverage, etc. High fidelity means that the selected data can fit the real data (e.g., having similar topologies, routing policies, traffic models, etc.), and high coverage means that the selected data can cover as many scenarios as possible.

6.1. Seed Sample Selection Phase

In the seed sample selection phase, high-quality seed samples are selected through the following steps to provide high-quality initial samples for the incremental optimization phase.

STEP 1: Training feature extraction model and feature extraction.

(1.1) The training data D’ is selected from the candidate data D according to the selection strategy. For the high fidelity strategy, the real data is used as the training data D’; for the high coverage strategy, the real data and simulated data are used together as the training data D’.

(1.2) Feature extraction model E is trained using the training data D’. Feature extraction model E is a network performance evaluation model that can be used to evaluate performance indicators such as delay, jitter and packet loss (such as RouteNet).

(1.3) Use the feature extraction model E obtained in STEP (1.2) to extract the feature of the training data D’ obtained in STEP (1.1). A network can be defined as a set of flow F, queue Q, and link L. The link state SF (such as link utilization), queue state SQ (such as port occupation), and flow state SL (such as delay, throughput, packet loss, etc.) are taken as features. Each sample in the training data D’ is converted to a feature vector [SF,SQ,SL].

STEP 2: Clustering.

Cluster the training data D’ after feature extraction. Clustering (such as K-means and DBSCAN) is an unsupervised machine learning technique that can automatically discover the natural groups in the data, divide the data into multiple clusters, and the samples in the same cluster have similarities.

Repeat STEP 3 and STEP 4 until all clusters have been traversed.

STEP 3: Calculating cluster centers and nearest neighbors.

(3.1) Calculate cluster centers. The method of calculating cluster centers is determined according to the clustering algorithm used in STEP 2. For example, using K-means clustering algorithm, the cluster center is calculated by finding the average of all data points in the cluster. These cluster centers are added to the seed dataset DS.

(3.2) Calculate k nearest neighbors of each cluster center and add them to the seed dataset DS. Suitable nearest neighbor calculation methods can be used, such as Euclidean distance, cosine distance, etc.

STEP 4: Expert knowledge verification.

(4.1) Expert knowledge can be used to verify the validity of samples through the range of indicators such as delay, queue occupation, and link utilization. If the verification passed, go to STEP 3. Otherwise, go to STEP (4.2).

(4.2) Randomly select m samples from the seed dataset DS and remove them. Calculate the nearest neighbors of the removed m samples, add them to the seed data set DS, and go to STEP (4.1).

6.2. Incremental Optimization Phase

The seed samples are taken as the initial training dataset. The filter model investigates the remaining candidate samples to filter out the easy, hard and OOD samples. Then the easy samples and hard samples are added to the training dataset. These processes are repeated to iteratively optimize the filter model and the training data until the high-quality data meets the constraints.

* Easy samples: Easy samples are data points where the model's predictions align closely with the true labels, often with high confidence. While training on easy samples can lead to good performance on familiar data, relying solely on them may limit the model's ability to handle complex or ambiguous cases, potentially causing overfitting and poor generalization to unseen data.

* Hard samples: Hard samples are data points where the model struggles, producing inaccurate, ambiguous, or low-confidence predictions. These samples are crucial for improving model robustness and generalization, as they expose weaknesses and encourage learning more discriminative features. Techniques like Online Hard Example Mining (OHEM), contrastive learning (focusing on hard negatives), and curriculum learning (gradually introducing harder samples) leverage hard samples to enhance model performance, prevent overfitting, and identify potential data issues such as labeling errors or biases.

* OOD samples: OOD samples refer to data points that significantly deviate from the training distribution, which should be detected and removed. Common detection methods include uncertainty estimation (e.g., Bayesian neural networks), density-based approaches (e.g., VAEs), distance-based metrics (e.g., Mahalanobis distance), outlier exposure, and energy-based models.

7. Use Cases

NDT can be applied to various types of networks, including data center networks, IP bearer networks, vehicular networks, wireless networks, optical networks, and IoT networks. This section highlights the significance of data generation and optimization in NDT by presenting several typical use cases.

7.1. Configuration Evaluation and Optimization in Data Center Networks

Data centers are essential for the growth of Internet services, consisting of numerous computing and storage nodes linked by a data center network (DCN), which serves as the communication backbone. The DCN faces challenges related to its large scale, diverse applications, high power density, and the need for reliability. NDT can evaluate configurations and technologies to reduce the risk of failures. For NDT to be effective, it must accurately model DCN traffic. A key challenge lies in generating realistic network traffic. By analyzing traffic patterns, data generation and optimization techniques can assist in creating simulated network data and optimize both real and simulated data. Numerous factors, such as the type of business, network size, volume of traffic, and load, influence traffic patterns in extensive DCNs. Moreover, these

traffic patterns are dynamic and evolve over time. For instance, workloads that are sensitive to latency, like online transaction processing, tend to peak during the day, whereas workloads for online analytical processing are more prevalent at night.

7.2. Performance Prediction in IP Bearer Networks

Internet service providers encounter challenges in delivering high-bandwidth, low-latency, and reliable services, especially in large networks like metropolitan area networks (MANs) . The widely adopted IP protocol adheres to a best-effort principle, making predictable performance difficult and complicating the stability and availability of network services during failures. NDT can function as a high-fidelity simulation platform for predicting IP bearer network performance. Accurate network status information is vital for optimizing protocols and identifying faults. Recent advancements in in-band network telemetry (INT) technology have allowed the integration of network performance data into packet headers on the data plane. Utilizing real performance data from INT, data generation and optimization techniques can create fine-grained simulated data, enhancing both real and simulated datasets for better model training outcomes.

7.3. Task Offloading in Vehicular Networks

The rise of vehicular networks has facilitated various delay-sensitive applications, including autonomous driving and navigation. However, vehicles with limited resources struggle to meet the low/ ultra-low latency requirements. To address this, computationally intensive tasks can be offloaded to resource-rich platforms like nearby vehicles, edge servers, and cloud servers. The dynamic nature of these networks, along with strict low-delay demands and large task data, presents significant offloading challenges. NDT is an emerging method that allows real-time monitoring of vehicular networks, aiding in effective offload decisions. Additionally, machine learning algorithms are increasingly utilized for task offloading to enhance accuracy and efficiency. Unlike traditional communication networks, vehicular networks are more dynamic and heterogeneous, leading to data shortages and quality issues. Data generation and optimization techniques can simulate data for adaptability and filter high-quality data from various sources, thereby improving model training effectiveness.

8. Discussion

Several topics related to data generation and optimization for NDT performance modeling require further discussion.

* Data generation methods: 1) Generate configurations that cover enough scenarios and scale from small to large networks. 2) Choose data generators that consider accuracy, speed, fidelity, etc. 3) Use data augmentation technology to expand the training data by using a small amount of practical data to generate similar data through prior knowledge.

* Data optimization methods: 1) Select data from multi-source candidate data, including hard sample mining, OOD detection, etc. 2) Verify whether the data quality meets the requirements.

* Deployment: 1) Time/space complexity and explainability of the data generation and optimization methods. 2) Provide feedback for data collection to form a closed loop.

9. Security Considerations

TBD

10. IANA Considerations

This document has no IANA actions.

11. Informative References

[Data-Centric-AI]
ACM Computing Surveys, "Data-centric Artificial Intelligence: A Survey", 2025.

[I-D.irtf-nmrg-network-digital-twin-arch]
"Network Digital Twin: Concepts and Reference Architecture", 2025, <https://datatracker.ietf.org/doc/draft-irtf-nmrg-network-digital-twin-arch/>.

[m3] ACM SIGCOMM 2024 Conference, "m3: Accurate Flow-Level Performance Estimation using Machine Learning", 2024.

[MimicNet] ACM SIGCOMM 2021 Conference, "MimicNet: Fast Performance Estimates for Data Center Networks with Machine Learning", 2021.

[RouteNet] IEEE/ACM Transactions on Networking, "RouteNet-Fermi: Network Modeling With Graph Neural Networks", 2023.

Acknowledgments

TODO acknowledge.

Authors' Addresses

Mei Li
China Mobile
Beijing
China
Email: limeiyjy@chinamobile.com

Cheng Zhou
China Mobile
Beijing
China
Email: zhouchengyjy@chinamobile.com

Danyang Chen
China Mobile
Beijing
China
Email: chendanyang@chinamobile.com

Qin Wu
Huawei
Email: bill.wu@huawei.com

Yuanyuan Yang
Huawei
Email: yangyuanyuan55@huawei.com