Sustainable Machine Intelligence Journal

Journal Homepage: sciencesforce.com/smij

Sustain. Mach. Intell. J. Vol. 8 (2024) 35–45

Paper Type: Original Article

GH-Twin: Graph Learning Empowered Hierarchical Digital Twin for Optimizing Self-Healing Networks

Nour Moustafa [1,*]

[1] School of Engineering and Information Technology, University of New South Wales@ ADFA, Canberra, ACT, 2600, Australia; nour.moustafa@unsw.edu.au.

Received: 31 Jan 2024 **Revised**: 07 May 2024 **Accepted**: 08 Jun 2024 **Published**: 10 Jun 2024

Abstract

Communication networks are witnessing a fast evolution towards Beyond 5G (B5G), bringing unprecedented complexities and challenges for optimizing networks in guaranteeing self-healing abilities and maintaining quality of services (QoS). To this end, this study presents a Graph Learning-driven Hierarchical Digital Twin framework, called GH-Twin, to build a reliable virtual replica of network components and their communications between different layers, leading to inclusive network representation. The proposed framework introduces graph cross-learning (GCL) distributed across different participants to devise competent predictive modelling of network performance collaboratively and preemptively recognize abnormalities in network settings. To preserve local privacy, differential privacy is applied by injecting some Gaussian into the parameters of local GCL before sharing it with the global coordinator. Proof of concept simulations has demonstrated that GH-Twin can precisely predict flow-level QoS and recognize anomalous links and nodes under different network topologies.

Keywords: Graph Learning; Hierachieral Digital Twins; Self-Healing Networks; Privacy-Preserving; Anomaly Detection.

1 | Introduction

The convergence of Fifth Generation (5G) and Internet of Things (IoT) technologies have been creating a flourishing of multitalented and extraordinary applications, including smart transportation, smart buildings, smart healthcare, autonomous driving/flight, and smart cities [1]. This envisaged that the network infrastructure would encounter various technical challenges and different quality-of-service (QoS) needs regarding computational resources, throughput, latency, and dependability [2]. The 5G and beyond 5G (B5G) wireless communications can support the requirements of IoT through dynamic network ecosystems. Nevertheless, B5G systems are known to require structural advancements to competently satisfy different QoS requirements on a joint network infrastructure [3].

Self-healing networks have arisen as a severe frontier in modern communication infrastructure, providing the autonomous ability to detect, diagnose, and mitigate network faults or disruptions without human intervention, ensuring nonstop operations and minimalizing downtimes. With the integration of Artificial Intelligence (AI) algorithms, self-healing mechanisms empower communication networks to adjust to varying scenarios, re-route traffic and enthusiastically assign resources to reserve optimal performance phases. In

Corresponding Author: nour.moustafa@unsw.edu.au
https://doi.org/10.61356/SMIJ.2024.8289

response to the growing complexity and scale of B5G networks, self-healing abilities are essential to address evolving communication challenges, including bottlenecks, equipment failures, cyber-attacks, etc. Intrinsically, self-healing networks come to be a foundation to achieve resilient and reliable communication infrastructures that meet the digital age's demands.

Digital Twinning is an evolving technology revolutionizing network management by providing a virtual replica of a physical network, which can improve predictive analytics and related optimization. With this ability to mirror real-world entities, the operators of self-healing networks can simulate, monitor, and analyze the behaviours of different network layers in a virtual environment, thereby offering preemptive intuitions about possible performance, failures, anomalies, and opportunities to optimize the network [4]. This, in turn, implies that the digital twin offers the operators of self-healing networks an advanced proactive decision-making capability that facilitates planning maintenance activities, optimizing resource allocation, and adjusting to network dynamics [5].

To design a digital twin framework, graph learning is a branch of AI algorithms that contributes to constructing reliable virtual replicas of complex infrastructures of self-healing networks. Graph learning can effectively represent network topologies, configurations, traffic flows, and QoS metrics in a structured and scalable fashion. This allows inherent interrelationships and interplay between different communication entities, which improve the fidelity and predictive proficiencies of the created virtual replicas. Additionally, it empowers self-healing networks with actionable intelligence for proactive management and optimization [6].

Several research studies have explored the potential of digital twinning for optimizing the performance of communication networks. However, many research gaps persist. Firstly, the current literature emphasized on the isolation aspects of network optimization and overlooked the multi-layer behaviour of network infrastructure. This highlighted the need to consider the interplay between physical attributes, network topology, and service-level necessities to enable self-healing services. Second, the limited scalability in large-scale networks persists as a significant obstacle to deployment in dynamic networks. This necessitates developing a novel learning approach to adapt to growing network dynamics while optimizing performance. Third, the practicality of digital twins for self-healing communications requires reliable, valid frameworks and testbeds for simulating real-world network scenarios.

To address these challenges, this work proposes a hierarchical digital twin framework, named GH-Twin, to contribute to the optimization of self-healing networks while satisfying the relevant QoS requirements of intelligent applications in an IoT environment. The contribution of this framework is as follows:

- A new GH-Twin framework involves central and edge digital twin components to provide dissimilar self-healing functionalities spanning from core network infrastructure to the network's periphery, promoting targeted optimizations to advance overall network performance in self-healing settings.
- GH-Twin introduces a novel privacy-preserving and scalable distributed graph cross-learning (GCL) technique to learn traffic information's inter-relations across different network layers and generate truthful predictions for the QoS metrics.
- Proof-of-concept simulations on different networking topologies validated the GH-Twin's efficiency for predictive QoS performance modelling.

The remainder of this article is summarized as follows: Section 2 covers the literature review. The system design is provided in Section 3. GH-Twin is described in section 4. The empirical simulations are discussed in section 5. Section 6 is provided in section 6. Finally, challenges and opportunities are discussed in section 7.

2 | Background and Literature

This section provides an in-depth overview of AI-empowered network optimization in a B5G communication environment. Many different types of IoT services and applications will need to be supported by B5G, and network optimization is seen as a crucial enabler for this. Thus, the recent research literature has devoted much effort to investigating the subject matter. For example, the work [7] presented a multi-domain orchestration architecture for network optimization by proposing the concept of a Multi-domain Service Conductor stratum to control services across several federated subdomains. This conductor layer examines receiving multi-domain slice queries and assigns them to the appropriate administrative layers based on the services those requests demand.

Additionally, a cross-domain coordinator was established to manage the dynamics associated with federated resource allocation effectively. In contrast, a cross-domain controller such as this one is responsible for managing the whole lifespan of a multi-domain slice and ensuring that all cloud and networking resources are coordinated across all unified subdomains. Recently, AI has been demonstrated to be an efficient tool for empowering network optimization and resource allocation in B5G communications. For example, the work [8] reviewed the components of network optimization technology in terms of data accumulating, storing data, data administration, and analytics and how these components relate to ideas of network optimization and the associated trade-offs. A comprehensive methodology was proposed for enacting SLA-accountable, big-data-driven dynamic slicing of available resources. For example, it is possible to use limited DL to create traffic forecasters, resource distribution mechanisms, and SLA implementation for low-complication slices.

In [9], the authors introduced a joint method for network optimization and routing optimization by integrating the network orchestration and control mechanism to facilitate fine-grained, instantaneous, and dynamic resource distribution. To address this resource allocation challenge, a Graph Convolutional Network (GCN) was developed to allow Multi-Task Deep Reinforcement Learning. The multitasking aspect was introduced to reinforcement learning by matching various output branches with joint scheduling resources across each network slice. A GCN equipped with a differentiable pooling layer is incorporated into a reinforcement learning model to capture topological data from graph-structured networks better. In another way, The work [10] presented a climbable digital twin of network optimization that seeks to learn the tangled relations among slices and examine the QoS metrics of slices under varied network settings using a simple graph network.

Furthermore, the authors [11] presented a new deep reinforcement learning method to deliver a federated and dynamic allocation and orchestration of network resources (i.e., spreading factor, transmission power, etc.) to satisfy various QoS metrics for industrial IoT systems. The proposed method consists of two primary steps. 1) a multiagent deep Q-learning optimization of dynamic resources that seeks to boost the self-QoS metrics.2) FL mechanism to train multiagent self-model in such a way that allows finding ideal allocation decision for spreading factor, transmission power for satisfying QoS reward for the logical slice. The study [12] presented a trustful and effective decentralized, federated slicing framework that systematically considers the design tenets and critical tasks in achieving blockchain-supported network optimization.

3 | System Design

In most cases, network operations are backed by a B5G substrate network, which makes a varied assortment of available resources. In GH-Twin, graph data is better exploited to represent the substrate networks, containing network edges and links connecting the nodes within the graph. Every real node has a certain number of network assets, including storage, graphics processing unit, central processing unit, and RAM. The resources of a given node are signified by employing a vector of resources, and various sorts of resources are epitomized in different ways to take the form of standard input. In this context, the proposed GH-Twin system treats every node as possessing the same number of resource categories, thereby having an identical size of resource vectors. If devices from various network domains have multiple categories of resources, each category is placed on a particular spot in the resources vector. For example, once a node is declared to belong to a specific network and isn't assigned any resources, the relevant spot is allotted a zero in the appropriate vector. The linking between any two nodes has a certain bandwidth across the substrate network. The self-healing network offers various IoT services, including intelligent transportation, smart finance, and smart agriculture, which significantly differ in the QoS metrics such as jittering, latency, throughput, etc.

GH-Twin is proposed to create a digital twin of slicing-enabled IoT networks to promote taking full advantage of the B5G features to satisfy each slice's QoS metrics [11]. This way, the proposed solution considers latency and jittering as QoS metrics since latency is one of the essential factors of IoT applications and constantly has a stringent necessity. The task of supplying an endwise network optimization of the digital twin network G^P necessitates an efficient orchestration of resources allocated to the nodes in every VNF and choosing a collection of applicable links to connects them in an efficient manner that enable satisfying the optimal resource distribution [13].

4 | Proposed GH-Twin Framework

This section first presents the formulation of the problem of end-to-end optimization of self-healing networks based on graph representation. Then, it discusses distributed graph learning as a part of the foundation of our hierarchal digital twin framework.

4.1 | Problem Formulation

The challenge is figuring out how to evaluate the QoS performance of self-healing network with multiple layers built on top of a shared physical infrastructure, as shown in Figure 1.

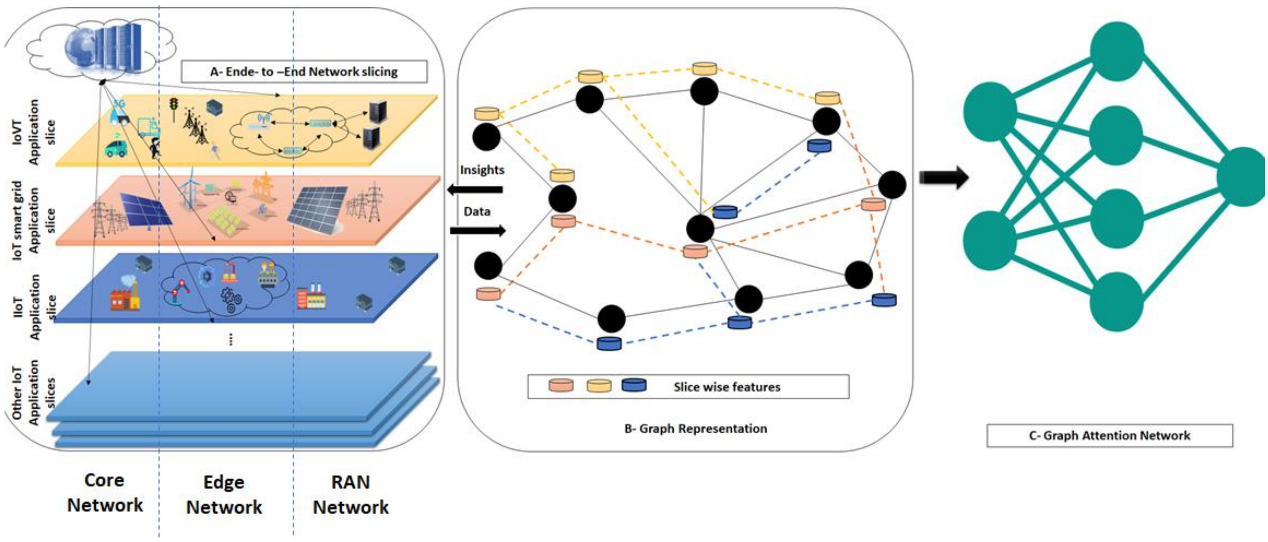


Figure 1. Visualization of graph representation for different layers in self-healing network.

Core networks, edge networks, and radio access networks are all part of the multi-layered architecture depicted in Figure 1 A). The resources are allocated according to various IoT services, necessitating executing certain operations in a specific order. Every network operation needs resources from various network domains. One way to visualize this sliced self-healing network infrastructure is shown in Figure 1 B), which uses a graph representation. Several virtual network functions (VNFs) from the same or different slices can be hosted on the same or different physical nodes in various domains. More than one virtual connection can use the same set of physical connections. At the same time, the immersive links can be spread across multiple physical ones. For example, virtual connections in the green, blue, and yellow slices are based on a web of actual interconnection. Therefore, the QoS metrics go beyond the standing of each linkage between these VNFs and instead focus on traffic patterns getting passed through an ordered collection of the VNFs.

The GCL algorithm is built on top of the graph representation as its fundamental building block. In most cases, the self-healing network that contains various resources and is depicted in Figure 1 A is exemplified in a multi-dimensional array. This ignores the possibility of collaboration between the various layers of the self-healing network as well as the connection between the network's topology and the resources that are present.

To circumvent the problems caused by these restrictions, a graph representation is used rather than a multi-dimensional matrix representation. This makes it possible to maintain intricate and vital connections. In this regard, the graph representation is formed with the nodes' statuses marked as hi to reflect the features of the slicing traffic that passes through the source vertex in addition to the usage ratio of links associated with this node (see Figure 1 B).

It is possible to determine the status of a particular layer by looking at the states of all the nodes that are contained inside it. This takes into account the fundamental characteristics that are required to calculate QoS metrics. After That, a decentralized and graph-based FL framework that effectively delivers honest slicing performance while having adequate elasticity to acclimatize to the dynamic conditions of network deployment, such as variation in the kind of physical topologies, variances in resource utilization rates, variation in relevant IoT services, and variation in the number of slices. In addition, given that the IoT network traffic has to go via a predefined order of configured VNFs within a specific part of a self-healing network, the packet losses and latency on any connection are examined because they are probable to affect the overall performance.

4.2 |Local Model

The pattern of attention procedures in the local model is rigorously inspired by Additive Attention; however, the model is unconvinced about the particular choice of attention system. The input of each layer is comprised of a set of features. The GCL layer generates an output consisting of a group of node features as the relevant outcome.

The construction of the attention mechanism in the local model could be broken down into four specific stages, with each level being characterized in the following manner. First, a linear transformation is to get an adequate degree of communicative capacity to convert the graph data from the input space to high-level feature spaces; it is necessary to have a set of one or more linear transformations that are amenable to learning. To accomplish this goal, the first step that must be taken in the GCL layer is to perform a joint linear transformation on each node while also applying a parameter matrix of weights to the transformation. Second, the attention factors are computed as paired non-normalized attention scores for each pair of neighbours. In particular, the embeddings of each couple of nodes, z, are combined, and the generated vectors are passed to dot product operations with the attention score vector. The combined embedding is non-linearly activated with $LeakyReLU$ activation. In common parlance, this type of attention is referred to as additive attention, as opposed to the dot-product attention illustrated by the Transformer network [14]. It is important to remember that the graph structure is included in the process through masked attention, whereby the parameters are calculated for all nodes in the first-level proximity of $i-th$ node within the graph. The third step, normalization, employs the SoftMax function to standardize parameters across all possible values of j, making them uniformly comparable across different nodes via the SoftMax function. Fourth, in the aggregation stage, as with GCN, the embeddings from different neighbours are combined while being mounted by the attention weights.

In a manner analogous to the multi-channel convolutions introduced in GCN, the GCL introduced the idea of using several observers to improve the model's performance and the consistency of its training. In such a scenario, numerous attention heads are created, each with unique settings. Each head can estimate its output simultaneously, with the combined output being the result of either concatenating or averaging the results. The concatenation operation is frequently applied in the intermediate attention layers, whilst the average operation is typically applied in the last attention layer.

4.3 |Distributed Learning

Figure 2 illustrates the procedure of federated training for the proposed GH-Twin framework, which passes through the following three phases:

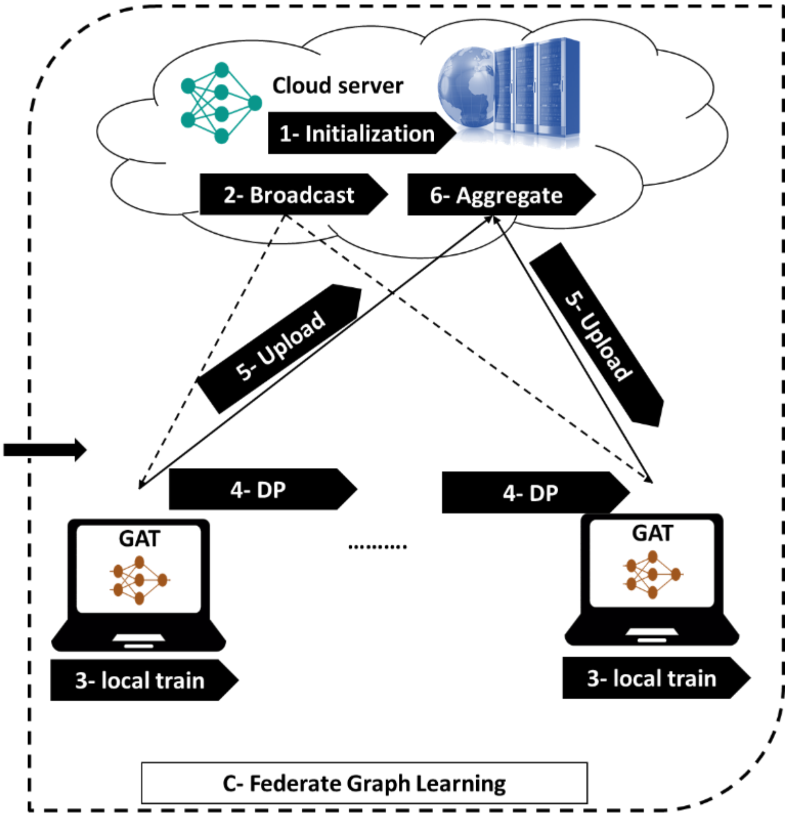
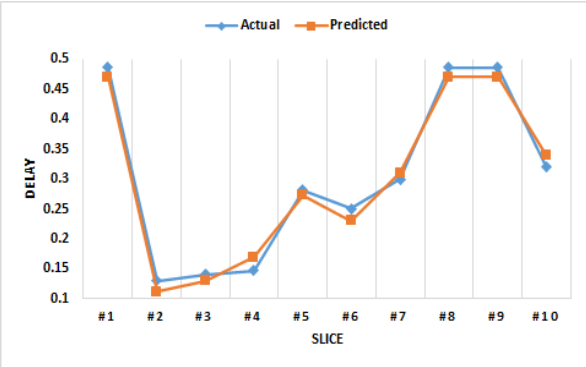


Figure 2. Illustration of the architecture of the proposed GH-Twin.

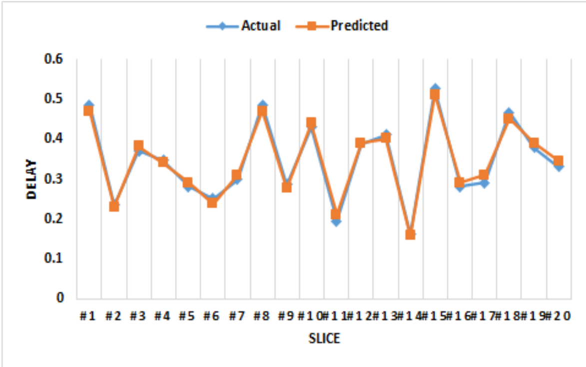
Step 1: Set initials: This step represents the starting point of the distributed learning of GCL, in which the initial parameters and structures are determined to build the distributed graph representation of the topology of the self-healing network. This involves defining nodes representing network entities (i.e., routers, switches, etc.) and edges denoting communication links. In addition, Domain knowledge-based pre-trained embeddings are used to initialize node and edge features, seizing semantic relations and contextual information intrinsic in a self-healing network. Besides, the hyperparameters of GCL and the training configurations (i.e., batch size, weight initialization, epochs, communication rounds, learning rate, etc.) are decided. Moreover, this step includes client selection, determining communication protocols, and aggregation algorithm.

Step 2: Train locally:

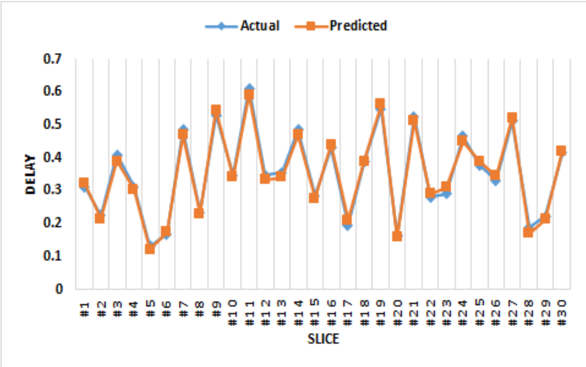
Herein, each selected client participates the task of local training of GCL using its exclusive training data. This decentralized allows collaborative modeling of QoS performance. In this step, each client optimizes the parameters of its GCL model via gradient-based optimization (Adam optimizer). The Log-Cosh loss is used as objective function measuring the difference between forecasted and actual QoS performance metrics detected in the client's data. Traditional training may suffer from data scarcity and distributional shift between clients, thereby, a graph perturbation is applied to insincerely present disparities into the training data, empowering the robustness of GCL model to varied networking conditions.



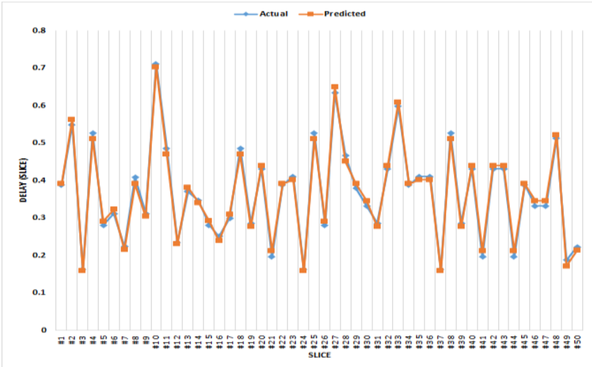
A) The NSFNet topology with 10 deployed slices



B) The GBN topology with 20 deployed slices



C) The GEANT2 topology with 30 deployed slices



D) The germany50 topology with 50 deployed slices

Figure 3. Illustration of the estimated delay compared to actual delay for each slice in different network topologies.

The adjusted parameters of each participant are transferred to the FL server for calculation of the global model parameters after each communication cycle between the member and the FL server. On the other hand, the information present in the GCL local parameters could be utilized by attackers to deduce the local slicing data that was employed for training. Therefore, maintaining the confidentiality of local characteristics becomes crucial to ensuring the GH-Twin's dependability.

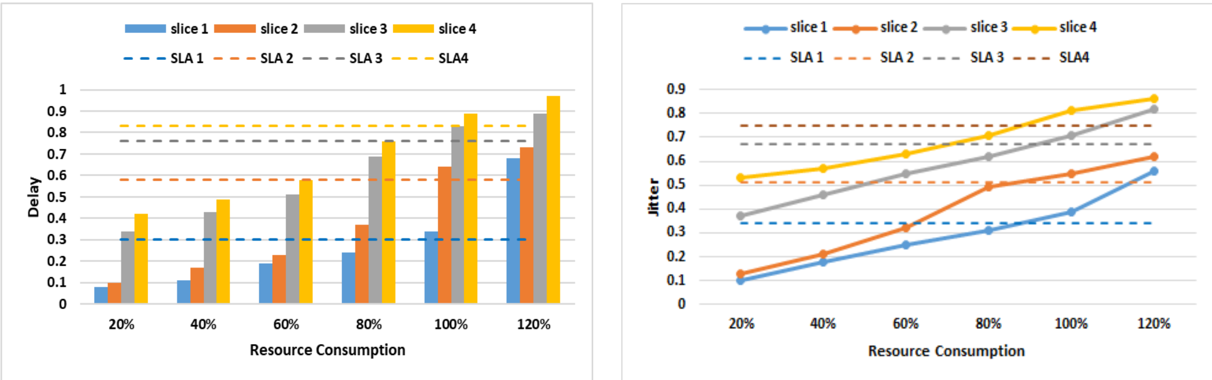
Earlier research looked at several approaches to solve this issue. Still, these methods are insufficient to handle resource-constrained and real-time IoT applications because they involve a lot of computing and communication latency. To address this problem, (ϵ, δ)-differential privacy (DP) mechanisms have been devised. These systems work by adding noise to local parameters and then pushing the results to the central aggregator. Consequently, before submission, the suggested GH-Twin uses DP to introduce Gaussian noise into the local GCL's parameters.

Step 3: global upgrade: Once the clients finish the local training of GCL, the training process shifts to the global update step, in which the gradients from all clients aggregate and build a global GCL model. With this cooperative process, the global model is guaranteed to collect intuitions from varied network sectors while conserving the privacy and security of training data. During this phase, the quantisation method minimises communication overhead to provide an easily compressed edition of local GCL updates before being uploaded for aggregation. This measure empowers the GH-Twin to be practicable and scalable enough for deployment in large-scale networks of B5G. The quantized model updates are aggregated with a secure averaging (safe-learn) algorithm based on multi-party computation (MPC) while keeping the local updates of GCL unexposed to either the central server or other clients.

5 | Experiments and Discussions

5.1 | Dataset Descriptions

Owing to the lack of a substantial dataset comprising the network performance data in the current literature, this study proposes to transform the four benchmarks engendered with a discrete event packet-level network called OMNet++ in Graph Neural Networking Challenge 2020 [15]. The generated data characterizes the paired source-destination traffic measurements through various routing paradigms, traffic samples of different slicing styles, and networking topologies. The selection of these datasets is not only due to their universal specifications but also because of per-source/destination values, which enable them to imitate the metrics of QoS performance. The data in the first benchmark are aggregated from NSFNET networking topology accompanied by 42 connections and 14 nodes. The second dataset belongs to the GBN networking topology with 52 connections and 17 nodes.



A) GH-Twin delay performances

B) GH-Twin jitter performance

Figure 4. Comparison of the performance of the GH-Twin against predefined SLA requirements under different resource consumption ratio.

The data in the third benchmark is accumulated from the GEANT2 networking topology, including 74 connections, and 24 nodes. In the fourth benchmark, the samples are synthetically engendered from germany50 networking topology comprising 276 connections and 50 nodes. The VNF features in each slice involve the amount of traffic synthesized and the number of packets sent with distinct resource consumption ratios. The ground truth label of each slice under predefined resource utilization includes delay, jittering, and packet loss.

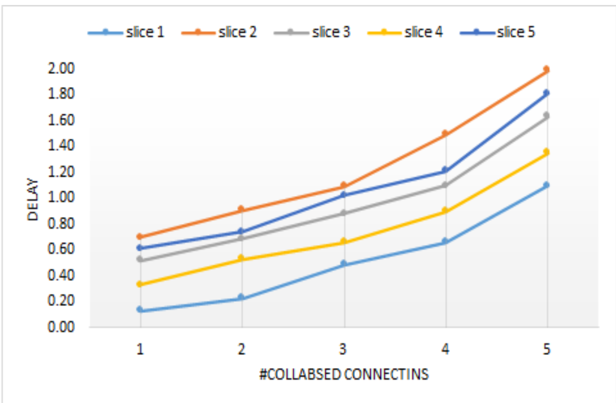


Figure 5. slice delay performance under different number of collapsed connections

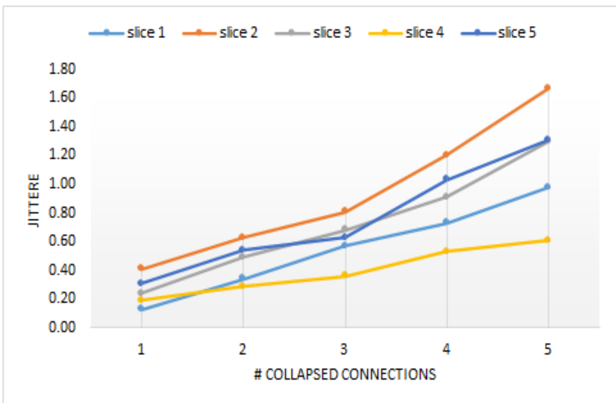


Figure 6. Slice jitter performance under different number of collapsed connections.

5.2 |Simulation Strategy

Simulation tests are carried out on a Dell workstation equipped with a 64-bit Windows 10 CPU (Intel (R) Xeon (R) CPU E5-2670 0@ 2.60GHz), 256GB of RAM, and an NVIDIA Quadro server GPU. The model implementation is done using the Pytorch library in a Python 3.7 environment. A dropout rate of 0.2 is chosen to prevent overfitting problems. Using a mini-batch size of 32 and a starting learning rate of 0.001, the model undergoes training for 500 steps. Twenty clients participated in the training.

5.3 |Results and Discussions

To evaluate the proposed GH-Twin's performance, each slice's actual delay is compared with the predicted delay values by the proposed model, as shown in Figure 3. The results predicted by the trained model are under four distinct network topologies. Each network topology experiment uses a different number of deployed network slices. It was clear that the anticipated delay values closely matched the labels based on reality for all four network architectures. *The* mean absolute percentage error (MAPE) of 1.6% is attained over the NSFNet topology, the prediction MAPE 2.1 % is attained over the GBN topology, the prediction MAPE 1.9% of GREANT2 is achieved over the topology, the prediction MAPE 3.3% is gained over the germany50 topology. This further indicates a low error rate, validating the efficiency and effectiveness of the proposed GH-Twin.

The GH-Twin is efficient enough to be applied as a reliable tool for inspecting the effect of significant metrics on the network slice's performance, which qualifies it to be employed in the management of self-healing networks. In this regard, three applications are possible for the GH-Twin in the context of network optimization tasks. The resource consumption in the given slice is active and changing over time. To optimise the self-healing network's performance, it should be aware of network optimization delay under various conditions of resource consumption in close real-time prior to making decisions.

In this way, the proposed GH-Twin is experimented to observe the delay/ hitter of a specific slice, while investigating whether the Service Level Agreement (SLA) prerequisites are breached when the resource consumption rises. A set of five slices is deployed along with delay and jitter requirements based on the GEANT2 topology, beginning with minimal resource consumption (20%). To experiment with circumstances of extreme network loads, slice and background traffic are increased by 20% per trial. The experiment was repeated six times, and the results are shown in Figures 4 (A) and B (B), which depict variations in slice delay and jittering versus various resource consumption rates. The GH-Twin can observe the SLA contravention by contrasting the forecasted slices' performance with a predetermined SLA, which can be applied to create an alert in case of an SLA violation.

Following this, to further validate the proposed GH-Twin's effectiveness, an additional experiment is performed to analyze the behaviour of the GH-Twin framework under the occurrence of extraordinary situations, i.e., connection malfunctions. When a specific running-on connection with the network slices collapses, it is essential to discover an alternative pathway to escape this collapsed connection and move around the influenced slices. In this experimentation, five slices with a low resource consumption rate are deployed in GBN topology, where a couple of slices share one or more physical connections. An arbitrary set of connections with deployed slices is eliminated, and then the shortest path routing algorithm is employed to locate an alternate route for the concerned slices. In each trial, we increase the number of removed connections by 1, starting from 1 and ending with five and evaluate the performance of the proposed GH-Twin by moving through the five network slices.

Figure 5 and Figure 6, respectively, present the delay and jittering performance of the proposed GH-Twin framework under a different number of collapsed connections. It is observable that increasing the number of collapsed connections increases the amount of delay or jitter owing to fewer accessible paths. The slices share more connections and occupy more resources. The GH-Twin framework can effectively deal with extraordinary situations and estimate the delay and jitter metrics for per-slice traffic movements, offering precious resource management and allocation intuitions.

6 | Conclusion

This work has studied the main challenges confronting the performance of self-healing networks in a spectrum of B5G communications. A hierarchal digital twin framework designed based on graph learning to demonstrate a reliable performance for network optimization. It encapsulates the GCL model to learn the logical structure of network slices locally to create a graph representation rather than transform the network into a matrix representation, then explicitly extract and learn insights from the graph representing the feature embeddings of the network slices. The local GCL collaboratively trained on the fog/edge nodes to optimize the global model in a privacy-preserved manner. The empirical evaluations demonstrate the effectiveness and efficiency of the GH-Twin to supervise the performance of network slices. Also, the experimental findings indicate that the GH-Twin is cost-valuable for overseeing SLA breakings, lightening the consequence of link collapses, and determining optimal network performance.

Acknowledgments

The author is grateful to the editorial and reviewers, as well as the correspondent author, who offered assistance in the form of advice, assessment, and checking during the study period.

Funding

This research was conducted without external funding support.

Data Availability

The datasets generated during and/or analyzed during the current study are not publicly available due to the privacy-preserving nature of the data but are available from the corresponding author upon reasonable request.

Conflicts of Interest

The author declares that there is no conflict of interest in the research.

Ethical Approval

This article does not contain any studies with human participants or animals performed by any of the authors.

References

[1] A. A. Barakabitze, A. Ahmad, R. Mijumbi, and A. Hines, "5G network slicing using SDN and NFV: A survey of taxonomy, architectures and future challenges," Comput. Networks, vol. 167, 2020, doi: 10.1016/j.comnet.2019.106984.

[2] M. Chahbar, G. Diaz, A. Dandoush, C. Cerin, and K. Ghoumid, "A Comprehensive Survey on the E2E 5G Network Slicing Model," IEEE Trans. Netw. Serv. Manag., vol. 18, no. 1, pp. 49–62, 2021, doi: 10.1109/TNSM.2020.3044626.

[3] X. Feng, J. Wu, Y. Wu, J. Li, and W. Yang, "Blockchain and digital twin empowered trustworthy self-healing for edge-AI enabled industrial Internet of things," Inf. Sci. (Ny)., vol. 642, p. 119169, 2023.

[4] H. B. Eldeeb, S. Naser, L. Bariah, S. Muhaidat, and M. Uysal, "Digital Twin-Assisted OWC: Towards Smart and Autonomous 6G Networks," IEEE Netw., p. 1, 2024, doi: 10.1109/MNET.2024.3374370.

[5] P. Yu et al., "Digital Twin Driven Service Self-Healing with Graph Neural Networks in 6G Edge Networks," IEEE J. Sel. Areas Commun., 2023.

[6] J. Lee, F. Solat, T. Y. Kim, and H. V. Poor, "Federated Learning-Empowered Mobile Network Management for 5G and Beyond Networks: From Access to Core," IEEE Commun. Surv. Tutorials, p. 1, 2024, doi: 10.1109/COMST.2024.3352910.

[7] T. Taleb, I. Afolabi, K. Samdanis, and F. Z. Yousaf, "On multi-domain network slicing orchestration architecture and federated resource control," IEEE Netw., 2019, doi: 10.1109/MNET.2018.1800267.

[8] H. Chergui and C. Verikoukis, "Big data for 5G intelligent network slicing management," IEEE Netw., 2020, doi: 10.1109/MNET.011.1900437.

[9] T. Dong et al., "Intelligent Joint Network Slicing and Routing via GCN-Powered Multi-Task Deep Reinforcement Learning," IEEE Trans. Cogn. Commun. Netw., 2022, doi: 10.1109/TCCN.2021.3136221.

[10] H. Wang, Y. Wu, G. Min, and W. Miao, "A Graph Neural Network-Based Digital Twin for Network Slicing Management," IEEE Trans. Ind. Informatics, 2022, doi: 10.1109/TII.2020.3047843.

[11] S. Messaoud, A. Bradai, O. Ben Ahmed, P. T. A. Quang, M. Atri, and M. S. Hossain, "Deep Federated Q-Learning-Based Network Slicing for Industrial IoT," IEEE Trans. Ind. Informatics, 2021, doi: 10.1109/TII.2020.3032165.

[12] Q. Hu, W. Wang, X. Bai, S. Jin, and T. Jiang, "Blockchain Enabled Federated Slicing for 5G Networks with AI Accelerated Optimization," IEEE Netw., vol. 34, no. 6, pp. 46–52, 2020, doi: 10.1109/MNET.021.1900653.

[13] A. De La Oliva et al., "5G-TRANSFORMER: Slicing and orchestrating transport networks for industry verticals," IEEE Commun. Mag., 2018, doi: 10.1109/MCOM.2018.1700990.

[14] A. Vaswani et al., "Attention is all you need," in Advances in Neural Information Processing Systems, 2017.

[15] K. Rusek, S. V. Jose, A. Mestres, B. R. Pere, and C. A. Albert, "Unveiling the potential of Graph Neural Networks for network modeling and optimization in SDN," in SOSR 2019 - Proceedings of the 2019 ACM Symposium on SDN Research, 2019, pp. 140–151. doi: 10.1145/3314148.3314357.