Graph Neural Network for Digital Twin Network: A Conceptual Framework.

Abubakar Isah
Department of ICT Convergence System Engineering
Chonnam National University
Gwangju, South Korea
abubakarisah@jnu.ac.kr

Hyeju Shin
Department of ICT Convergence System Engineering
Chonnam National University
Gwangju, South Korea
sinhye102@gmail.com

Ibrahim Aliyu
Department of ICT Convergence System Engineering
Chonnam National University
Gwangju, South Korea
aliyu@jnu.ac.kr

Rashid Muhammad Sulaiman
Department of ICT Convergence System Engineering
Chonnam National University
Gwangju, South Korea
msulaimanrashid@jnu.ac.kr

Jinsul Kim
Department of ICT Convergence System Engineering
Chonnam National University
Gwangju, South Korea
jsworld@jnu.ac.kr

***Abstract*— Graph Neural Networks (GNNs) have emerged as a powerful framework for analyzing and extracting information from complex network data. In the realm of Digital Twin Networks (DTN), where physical entities are mirrored in a virtual environment, GNNs offer a transformative approach by leveraging the inherent structure and relationships within digital twins. GNNs enable enhanced data representation and predictive modeling. DTNs encompass several core elements that naturally conform to a graph-like structure, including aspects like network topology and routing patterns. In this paper, we review the concept of graph neural network models, network of digital twin applications, and their comparison with other different fields.**

Keywords*— *graph neural network, digital twin networks, node-level, graph-level, edge level

I. Introduction

The rise of immersive services, such as virtual reality, augmented reality, holographic content, and metaverse development, is leading to increased complexity in communication networks [1]. To efficiently optimize and manage these intricate networks, a technology known as the digital twin network (DTN) or network digital twin (NDT) has emerged. This technology combines the capabilities of the digital twin (DT) with communication networks and offers promising solutions for intelligent network management [2]. The significance of DT technology has grown significantly across various domains, aligning with advancements in simulation and computing technology [3].

In the past, network models were typically constructed using analytical methods, predominantly relying on simulations or queueing theory. Nevertheless, these approaches have inherent limitations when it comes to emulating the complex behaviors of actual networks, such as handling the dynamics of real traffic, topologies, and complex routing scenarios [4]. Conversely, precise alternatives based on discrete-event simulation (for instance, tools like ns-3 or OMNeT++) struggle to efficiently handle extensive network environments. Their substantial computational demands not only restrict the ability to simulate large-scale real-world networks but also hinder their use for rapid-timeframe operations like online optimization.

Recent advancements in artificial intelligence offer a promising avenue for the development of network digital twins (NDT). The rise of data-driven machine learning techniques, particularly deep learning [5], has gained significant attraction in the field of networking. This has given rise to a new generation of models that can learn from data rather than relying on explicit programming. Researchers are increasingly utilizing deep neural networks (DNNs) to capture complex network behaviors [6] and devise decision-making strategies using deep reinforcement learning (DRL) [7]. Many of these efforts are built upon well-established learning architectures like convolutional neural networks [8], recurrent neural networks, and AutoEncoders [9], along with their variations.

However, it's important to note that communication networks are fundamentally graph-based, and most of the existing learning architectures aren't inherently designed to handle information structured in a non-Euclidean domain [10]. This limitation arises from the irregular topology of graphs and the interdependence between nodes. Consequently, these models face challenges in delivering accurate results when dealing with graph data and struggle to generalize across dynamic network topologies and configurations.

In our paper, we delve into the examine of Graph Neural Network (GNN) models that have emerged in recent years and explore the advancements in their utilization within Digital Twin Networks (DTNs) based on message-passing neural network (MPNN). Furthermore, we outline potential research directions that GNNs can pave the way for in the future of this field. These discoveries offer valuable insights for researchers in the realm of developing GNN-based DTNs. This review article underscores several key contributions:

- We provide a comprehensive overview of GNN models and their application in DTNs.
- We discuss the challenges and opportunities of using GNNs for DTNs.
- We identify promising research directions for GNN-based DTNs.

II. GRAPH NEURAL NETWORK

A. Basic Principles

The concept of graph neural networks was first introduced in [11] and subsequently expanded upon in [12] and [13]. These initial investigations belong to the class of recurrent graph neural networks (RecGNNs). They acquire the representation of a target node by iteratively transmitting information from its neighbors until a stable, unchanging state is achieved. This procedure is computationally demanding, and there have been growing endeavors to address these complexities, as evidenced by recent work in [14].

GNNs are a class of deep neural networks designed specifically for the analysis of data organized in a graph structure. A Graph can be represented as a tuple $\boldsymbol{G} = (\boldsymbol{V}, \boldsymbol{E})$, where V is the set of vertices (nodes) in the graph. $\boldsymbol{E}$ is the set of edges (links) in the graph. An edge $\boldsymbol{e_{ij}}$ is a directed edge from node V_i to node V_j. The neighborhood of a node V is the set of nodes that are directly connected to $\boldsymbol{V}$. The adjacency matrix A of a graph is a square matrix where each entry $\boldsymbol{A_{ij}}$ represents the presence or absence of an edge from node V_i to node V_j. If there is an edge from $V_i\ to\ V_j$, then $A_{ij} = 1$. Otherwise, $A_{ij} = 0$. A graph may include additional information in the form of node attributes denoted as X, in which X is a node feature matrix with dimensions $\mathbb{R}^{nxd}$, where X_v represents the feature vector of node $V \in R^d$. Additionally, the graph may incorporate edge attributes denoted as X^e. X^e is an edge feature matrix with dimensions R^{mxc}, where $X^e_{u,v}$ represents the feature vector associated with the edge $(v, u) \in \mathbb{R}^c$.

In cases of undirected edges, it's presumed that the weights associated with the edges are symmetrical, affecting both connected nodes equally. In other way rounf, directed graphs consist of edges that originate from one node and terminate at another. A spatial-temporal graph represents a unique scenario where the characteristics of nodes change over time. This can be expressed as $G(t) = (V, E, X(t), X^e(t)$, indicating that both node attributes and edge attributes are subject to temporal variations.

B. Message-Passing based on GNN

Message passing neural networks (MPNNs) are specifically designed to handle data that's organized as graphs. In these networks, nodes within a graph communicate with their nearby neighbors by exchanging messages. This process enables the network to learn and model the various connections and interactions between different elements in the graph. MPNNs have found widespread application in diverse fields, including predicting molecular properties, analyzing social networks, in recommender systems, and now in communications networks.

These applications share a common characteristic. The data they involve can be naturally represented as graphs, making MPNNs a powerful tool for understanding and working with such data. For instance, Table 1 represent the applications of GNN in different areas and in networking in which [15] proposed a message passing neural network to capture the structural designed where the smaller components can be defined as function:

$$h_u^t = Update^t\left(h_u^{(t-1)}, Agg^t\left(\left\{h_v^{(t-1)}, \forall v \in N(u)\right\}\right)\right) \quad (1)$$

$$= Update^{(t)}\left(h_v^{(t-1)}, m_u^{(t-1)}\right) \quad (2)$$

The two equations defined the message passing where h_u^t is the hidden state with embedding node $u, v \in V$ before being passed to temporal layet t in which the initial node features is $h_u^{(0)}$. There are two essential features ($Update^{(t)} and\ Agg^{(t)}$). Once the message $m_u^{(t-1)}$ was passed from the neighboring node u, the the hidden of node u will be updated with the previous node embedding and the aggregated feature was obtained.

TABLE 1. Models comparison and their applications

Model	Year	Applications	Metrics	Task
Directional-MPNN[16]	2019	Molecular properties	ROC, AUC, RMSE, MAE	Regression
Message-passing attention network[15]	2020	Chemical property prediction	ROC-AUC	Classification
MPNN extended With predictive Uncertainty[17]	2021	Molecular formation energies prediction	MAE and RMSE	Classification
Dual-branched MPNN[18]	2022	Molecular property prediction	MAE	Regression
RouteNet – Fermi MPNN[19]	2023	Network traffic prediction	MAPE	Classification

C. Recurrent Graph Neural Networks

Recurrent graph neural networks (RecGNNs) are foundational contributions to the field of GNNs. They employ a consistent set of parameters in a recurring manner across the nodes within a graph to derive more abstract node representations. Due to limitations in computational capabilities, early studies primarily concentrated on directed acyclic graphs, as indicated in [20] and [21]. These RecGNNs are designed to acquire node representations using recurrent neural structures. They operate under the assumption that a node within a graph continuously shares information with its neighboring nodes until a stable and unchanging state is achieved [9]. The conceptual significance of RecGNNs has had a profound impact on subsequent research in the realm of convolutional graph neural networks. In particular, the concept of message passing introduced by RecGNNs has been carried forward into spatial-based convolutional graph neural networks [22].

D. Convolutional GNNs

This category of Graph Neural Networks (GNNs), which is widely used in communication network, focuses on extending the concept of convolution from grid data to data structured as graphs [12]. The fundamental concept involves generating node representations by aggregating features from the target node and its neighboring nodes. Convolutional GNNs can be broadly classified into two groups: spectral-based methods, which are based on spectral graph theory and operate on the entire graph simultaneously, and spatial-based methods, which perform convolution operations only on a subset of nodes considered as neighbors of the target node. Spatial-based methods are more efficient because they don't require analyzing the entire graph at once, making them suitable for large-scale problems and real-time applications.

Additionally, for training approaches; a variety of GNNs, such as Convolutional GNNs (ConvGNNs), can undergo training either in a supervised or unsupervised manner [20], depending on the specific learning tasks and the availability of label information. This training occurs within a comprehensive end-to-end learning framework.

III. CONCEPTUAL FRAMEWORK

Utilizing both the graph structure and DTNs' real network data as inputs, GNNs offer diverse functionalities, allowing them to address various graph analytics tasks. Here are some specific examples of how GNNs with different structures can be used in networks of digital twins.

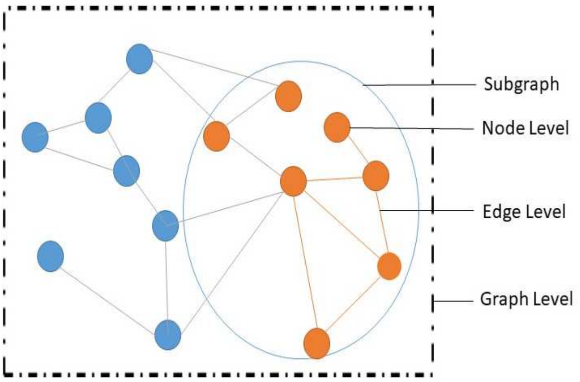

Fig. 1. A Graph Representation of GNN

A. Node-Level

Given a graph G = (V, E), the goal of node level learning is to learn the features of the targeted nodes in the graph v ∈ V. This includes node classification, node regression, and node clustering. Node classification tries to categorize nodes into several classes, and node regression [23]**.** At the node level, the outputs of GNNs are involved in tasks such as node regression and node classification. Both RecGNNs [20] and ConvGNNs [24] are capable of deriving advanced node representations through information propagation or graph convolution. By employing a multi-perceptron or softmax layer as the output layer, GNNs can seamlessly handle node-level tasks in an end-to-end fashion.

- **Node classification**: A node-level GNN can be used to classify the nodes in a digital twin network into different categories, such as machines, sensors, and products. This can be used to identify the different types of entities in the network and to understand their relationships with each other.
- **Link prediction**: A node-level GNN can be used to predict the presence or absence of links between nodes in a digital twin network. This can be used to identify potential new connections between entities in the network or to detect broken or faulty connections.
- **Anomaly detection:** A node-level or graph-level GNN can be used to detect anomalies in the behavior of a digital twin network. This can be used to identify potential problems with the network early on and to take corrective action before the problems cause any disruption.

B. Graph-level

At the graph level, the outputs of GNNs are associated with the task of graph classification. To achieve a concise representation at the graph level, GNNs are frequently integrated with pooling and readout operations.

- **Graph classification**: A graph-level GNN can be used to classify digital twin networks into different categories, such as manufacturing networks, transportation networks, and communication networks. This can be used to identify the different types of networks that are in operation and to understand their unique characteristics.
- **Graph regression**: A graph-level GNN can be used to predict the performance of a digital twin network under different conditions. This can be used to optimize the network's configuration or to identify potential bottlenecks.

It is also important to note that GNNs can be used in conjunction with other machine-learning techniques to fine tune and understand the complex structure of the graph. For example, we can use a GNN to learn to represent the nodes and edges in a digital twin network, and then use a traditional machine learning classifier to classify the nodes or predict the performance of the network.

C. Edge-level

Given a graph G = (V, E), the goal of edge level tasks is to apply the learning task on each edge $e \text{ € } E$. Edge-level tasks generally include edge classification/regression and link prediction; the former requires the model to classify edge types or predict properties for edges of the graph, and the latter tries to predict whether there is an edge between two given nodes [8]. Link prediction (also known as relation prediction and graph completion) aims to predict the unknown or missing relations/links between nodes in the graph, given an incomplete set of links/relationships between a training set of nodes $E_{train}\text{€ } E$.

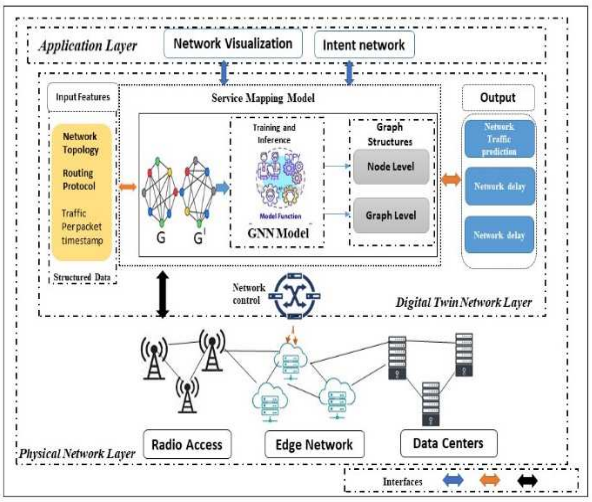

Fig. 2. A GNN-based Digital Twin Network Framework

D. *Application Network Layer*

The network application layer is a critical component of GNN-based digital twin networks. It allows external applications to interact with the DTN and leverage its GNN capabilities to solve complex problems. For example, external applications can use the network application layer to:

- Query the digital twin network for insights about the real-world system it represents. For example, an external application could query the digital twin network to identify potential bottlenecks in a manufacturing process or to predict the impact of a change in the environment on the performance of a transportation network.

E. *Network of Digital Twin Layer*

In a GNN-based DTN, the twin network layer plays a vital role in enabling the GNN models to learn from and operate on the DTN data. The twin network layer is essential for enabling GNN-based DTNs to provide valuable insights and predictions about the real-world systems they represent. By managing and organizing the DTN data and providing essential capabilities such as model topology management, security management, service mapping, and data storage, the twin network layer [25] enables the GNN models to reach their full potential. The twin network layer manages the topology of the digital twin network, including the nodes, edges, and their attributes. This information is essential for the GNN models to learn the relationships between the different entities in the DTN.

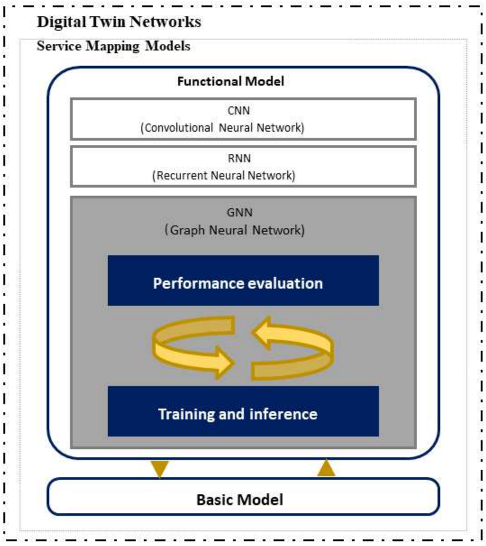

Fig. 3. A DTN Service Mapping Models

F. *Physical Network Layer*

The physical network layer is essential for enabling GNN-based DTNs to provide real-time insights and predictions about the physical networks they represent. By providing the digital twin network with access to the physical network data and control capabilities, the physical network layer enables the GNN models to learn the patterns of the physical network and to make informed predictions about its behavior. For instance, the physical network layer can be used to collect data from the servers and switches in the data center network. This data can be sent to the DTN, where it can be used to train GNN models to identify bottlenecks and inefficiencies in the network. The GNN models can then be used to recommend ways to improve the performance of the data center network.

IV. Research Directions

The future of research on digital twin networks for the GNN model using real network data based on understanding the structural graph analysis is very promising. GNNs are powerful tool for modeling and analyzing graph data, and digital twin networks are a valuable tool for predicting and managing the behavior of real-world systems. By combining these two technologies, we can create new and innovative ways to understand and manage complex networks.

A. *New GNN Architecture.*

- Developing new GNN architectures that are better at capturing the complex relationships between nodes in a digital twin network. For example, we could develop GNNs that are able to model the temporal dynamics of the network or the spatial relationships between nodes.

B. *Real-Time Network Data.*

- Using GNNs to learn new insights about the behavior of real-world systems. For example, we could use GNNs to identify the factors that contribute to network anomalies or to predict the impact of changes to the network on its performance.

C. *New applications for DTN*

- Developing new applications for digital twin networks based on node-level and graph-level analysis. For example, we could develop new tools for network planning, optimization, and troubleshooting.

D. *Research projects that are being pursued in this area.*

- **Researchers at Universitat Politècnica de Catalunya (Barcelona Network Center)** are working toward developing new GNN architecture for network optimization and control, especially the network of digital twins using real network data.

- **Researchers at the University of California**, Berkeley are developing a GNN-based system for predicting the performance of computer networks. This system could be used to identify potential bottlenecks in the network and to optimize the network's configuration.

- **Researchers at the Massachusetts Institute of Technology** are developing a GNN-based system for detecting anomalies in the behavior of industrial control systems. This system could be used to identify potential cybersecurity threats or to predict equipment failures.

- **Researchers at Siemens** are developing a GNN-based system for optimizing the production schedule of a manufacturing plant. This system could be used to reduce production costs and improve product quality.

V. Conclusion

GNN-based digital twin networks are a promising new technology with the potential to revolutionize the way we manage and optimize complex real-world systems. With graph neural networks, node-level and graph-level classification can combine very well to powerful learning capabilities in networks of digital twins. GNN can be trained to learn patterns and the behavior of the network to forecast delay that are likely to occur.

VI. Acknowledgement

This work was supported by Electronics and Telecommunications Research Institute(ETRI) grant funded by ICT R&D program of MSIT/IITP[2019-0-00260, Hyper-Connected Common Networking Service Research Infrastructure Testbed] and supported by the MSIT(Ministry of Science and ICT), Korea, under the Innovative Human Resource Development for Local Intellectualization support program(IITP-2023-RS-2022-00156287) supervised by the IITP(Institute for Information & communications Technology Planning & Evaluation).

References

[1] A. Isah, I. Aliyu, J. Park, and J. Kim, "Network Traffic Prediction Model in a Data-Driven Digital," pp. 15–19, 2023.

[2] A. Isah *et al.*, "A Data-Driven Digital Twin Network Architecture in the Industrial Internet of Things (IIoT) Applications," vol. 6, no. 2.

[3] Y. Yang, Y. Wei, and T. Shen, "A Review of Graph Neural Networks for Recommender Applications," *Proc. 2021 IEEE Int. Conf. Unmanned Syst. ICUS 2021*, pp. 602–607, 2021, doi: 10.1109/ICUS52573.2021.9641274.

[4] J. Suarez-Varela *et al.*, "Graph Neural Networks for Communication Networks: Context, Use Cases and Opportunities," *IEEE Netw.*, no. June, pp. 146–153, 2022, doi: 10.1109/MNET.123.2100773.

[5] M. Xia, H. Shao, D. Williams, S. Lu, L. Shu, and C. W. de Silva, "Intelligent fault diagnosis of machinery using digital twin-assisted deep transfer learning," *Reliab. Eng. Syst. Saf.*, vol. 215, no. June, p. 107938, 2021, doi: 10.1016/j.ress.2021.107938.

[6] K. Rusek, J. Suarez-Varela, P. Almasan, P. Barlet-Ros, and A. Cabellos-Aparicio, "RouteNet: Leveraging Graph Neural Networks for Network Modeling and Optimization in SDN," *IEEE J. Sel. Areas Commun.*, vol. 38, no. 10, pp. 2260–2270, 2020, doi: 10.1109/JSAC.2020.3000405.

[7] L. Wen, X. Li, L. Gao, and Y. Zhang, "A New Convolutional Neural Network-Based Data-Driven Fault Diagnosis Method," *IEEE Trans. Ind. Electron.*, vol. 65, no. 7, pp. 5990–5998, 2018, doi: 10.1109/TIE.2017.2774777.

[8] Y. Xu, J. Wang, M. Guang, C. Yan, and C. Jiang, "Multistructure Graph Classification Method with Attention-Based Pooling," *IEEE Trans. Comput. Soc. Syst.*, vol. 10, no. 2, pp. 602–613, 2023, doi: 10.1109/TCSS.2022.3169219.

[9] Z. Wu, S. Pan, F. Chen, G. Long, C. Zhang, and P. S. Yu, "A Comprehensive Survey on Graph Neural Networks," *IEEE Trans. Neural Networks Learn. Syst.*, vol. 32, no. 1, pp. 4–24, 2021, doi: 10.1109/TNNLS.2020.2978386.

[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*, vol. 18, no. 2, pp. 1367–1376, 2022, doi: 10.1109/TII.2020.3047843.

[11] F. Scarselli, M. Gori, A. C. Tsoi, M. Hagenbuchner, and G. Monfardini, "The graph neural network model," *IEEE Trans. Neural Networks*, vol. 20, no. 1, pp. 61–80, 2009, doi: 10.1109/TNN.2008.2005605.

[12] F. Scarselli, M. Gori, A. C. Tsoi, M. Hagenbuchner, and G. Monfardini, "Computational capabilities of graph neural networks," *IEEE Trans. Neural Networks*, vol. 20, no. 1, pp. 81–102, 2009, doi: 10.1109/TNN.2008.2005141.

[13] Z. Zhao, J. Qiao, J. Li, M. Shi, and X. Wang, "Distribution Network Topology Identification with Graph Transformer Neural Network," *2022 4th Int. Conf. Smart Power Internet Energy Syst. SPIES 2022*, pp. 1580–1585, 2022, doi: 10.1109/SPIES55999.2022.10082563.

[14] Y. Li, R. Zemel, M. Brockschmidt, and D. Tarlow, "Gated graph sequence neural networks," *4th Int. Conf. Learn. Represent. ICLR 2016 - Conf. Track Proc.*, 2016.

[15] J. Jo, B. Kwak, H. S. Choi, and S. Yoon, "The message passing neural networks for chemical property prediction on SMILES," *Methods*, vol. 179, no. April 2020, pp. 65–72, 2020, doi: 10.1016/j.ymeth.2020.05.009.

[16] K. Yang *et al.*, "Analyzing Learned Molecular Representations for Property Prediction," *J. Chem. Inf. Model.*, vol. 59, no. 8, pp. 3370–3388, 2019, doi: 10.1021/acs.jcim.9b00237.

[17] J. Busk, P. B. Jørgensen, A. Bhowmik, M. N. Schmidt, O. Winther, and T. Vegge, "Calibrated uncertainty for molecular property prediction using ensembles of message passing neural networks," *Mach. Learn. Sci. Technol.*, vol. 3, no. 1, 2022, doi: 10.1088/2632-2153/ac3eb3.

[18] J. Jo, B. Kwak, B. Lee, and S. Yoon, "Flexible Dual-Branched Message-Passing Neural Network for a Molecular Property Prediction," *ACS Omega*, vol. 7, no. 5, pp. 4234–4244, 2022, doi: 10.1021/acsomega.1c05877.

[19] M. Ferriol-Galmes *et al.*, "RouteNet-Fermi: Network Modeling With Graph Neural Networks," *IEEE/ACM Trans. Netw.*, vol. 2023, pp. 1–15, 2023, doi: 10.1109/TNET.2023.3269983.

[20] A. Sperduti and A. Starita, "Supervised neural networks for the classification of structures," *IEEE Trans. Neural Networks*, vol. 8, no. 3, pp. 714–735, 1997, doi: 10.1109/72.572108.

[21] A. Micheli, D. Sona, and A. Sperduti, "Recursive Cascade Correlation," vol. 15, no. 6, pp. 1396–1410, 2004.

[22] M. Ferriol-Galmés *et al.*, "Building a Digital Twin for network optimization using Graph Neural Networks," *Comput. Networks*, vol. 217, no. September, p. 109329, 2022, doi: 10.1016/j.comnet.2022.109329.

[23] S. Rahmani, A. Baghbani, N. Bouguila, and Z. Patterson, "Graph Neural Networks for Intelligent Transportation Systems: A Survey," *IEEE Trans. Intell. Transp. Syst.*, vol. 24, no. 8, pp. 8846–8885, 2023, doi: 10.1109/TITS.2023.3257759.

[24] Y. Tang, Z. Huang, J. Cheng, G. Zhou, S. Feng, and H. Zheng, "Graph Neural Network-based Node Classification with Hard Sample Strategy," *2021 Int. Conf. Cyber-Physical Soc. Intell. ICCSI 2021*, pp. 1–4, 2021, doi: 10.1109/ICCSI53130.2021.9736175.

[25] L. Hui, M. Wang, L. Zhang, L. Lu, and Y. Cui, "Digital Twin for Networking: A Data-driven Performance Modeling Perspective," *IEEE Netw.*, 2022, doi: 10.1109/MNET.119.2200080.