

Review

Empowering Digital Twin for Future Networks with Graph Neural Networks: Overview, Enabling Technologies, Challenges, and Opportunities

Duc-Thinh Ngo ^{1,2,*} , Ons Aouedi ², Kandaraj Piamrat ^{2,*} , Thomas Hassan ¹  and Philippe Raipin-Parvédy ¹ 

¹ Orange Innovation, 35510 Cesson-Sévigné, France; thomas.hassan@orange.com (T.H.); philippe.raipin@orange.com (P.R.-P.)

² IMT Atlantique, Nantes University, École Centrale Nantes, CNRS, INRIA, LS2N, UMR 6004, 44000 Nantes, France; ons.aouedi@imt-atlantique.fr

* Correspondence: duchinh.ngo@orange.com (D.-T.N.); kandaraj.piamrat@univ-nantes.fr (K.P.)

Abstract: As the complexity and scale of modern networks continue to grow, the need for efficient, secure management, and optimization becomes increasingly vital. Digital twin (DT) technology has emerged as a promising approach to address these challenges by providing a virtual representation of the physical network, enabling analysis, diagnosis, emulation, and control. The emergence of Software-defined network (SDN) has facilitated a holistic view of the network topology, enabling the use of Graph neural network (GNN) as a data-driven technique to solve diverse problems in future networks. This survey explores the intersection of GNNs and Network digital twins (NDTs), providing an overview of their applications, enabling technologies, challenges, and opportunities. We discuss how GNNs and NDTs can be leveraged to improve network performance, optimize routing, enable network slicing, and enhance security in future networks. Additionally, we highlight certain advantages of incorporating GNNs into NDTs and present two case studies. Finally, we address the key challenges and promising directions in the field, aiming to inspire further advancements and foster innovation in GNN-based NDTs for future networks.



Citation: Ngo, D.-T.; Aouedi, O.; Piamrat, K.; Hassan, T.; Raipin-Parvédy, P. Empowering Digital Twin for Future Networks with Graph Neural Networks: Overview, Enabling Technologies, Challenges, and Opportunities. *Future Internet* **2023**, *15*, 377. <https://doi.org/10.3390/fi15120377>

Academic Editor: Dimitris Mourtzis

Received: 1 October 2023

Revised: 11 November 2023

Accepted: 20 November 2023

Published: 24 November 2023

Keywords: communication network; digital twin; Graph neural network; deep learning

1. Introduction

The emergence of 5G networks holds the potential to reshape industries, moving beyond the revolution of mobile communication to provide instantaneous connectivity for billions of Internet of things (IoT) devices, fueling the digital transformation of Industry 4.0. As networking technologies continue to evolve in complexity and scale, cutting-edge paradigms such as Software-defined network (SDN), Network function virtualization (NFV), and network slicing offer exceptional flexibility and programmability in network management and service delivery. However, the deployment and optimization of these advanced networking paradigms present new complexities and challenges that require innovative solutions.

To address these challenges and enhance the capabilities of future networks, the concept of Network digital twin (NDT) has gained prominence [1–3]. Digital twin often refers to the digital replication of a physical object enabled by a two-way synchronization. With the advantage of modeling accurately complex dynamic systems, digital twin coupled with data-driven methodologies emerges as a technology to enhance performance, to enable smart monitoring, low-cost trials, and predictive maintenance on any complex system. This includes the next-generation network. Aiming at the accurate data-driven network models operating in real time, the leverage of Machine learning and Artificial intelligence (ML/AI) techniques inside an NDT is inevitable. Moreover, among the diverse bodies of the ML/AI literature, Graph neural network (GNN) is naturally one of the most suitable to



Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

model networking systems. GNN excels in processing graph-structured data where nodes represent elements and edges depict relationships. By effectively propagating information across interconnected nodes, GNN captures both local and global insights, making them particularly potent in addressing the intricacies of network behavior [4].

In this paper, we explore the potential of GNNs in empowering NDTs to address these challenges and drive the advancement of future networks. Through a comprehensive analysis of state-of-the-art research, this paper aims to contribute to the understanding and adoption of GNN-based NDTs as powerful tools for future network management and evolution.

1.1. Existing Reviews

To this end, several existing surveys have been dedicated to reviewing the various applications, development, and deployment of Digital twins (DTs) and GNNs within next-generation networks. However, these surveys typically address the two topics separately.

On one hand, Vesselinova et al. [5] reviewed several combinatorial optimization problems on graphs while having a special eye on the applications in the telecommunications domain. The combinatorial optimization problems discussed in the paper include tasks such as network design, routing, bandwidth allocation, scheduling, and network reliability. While this survey does not explicitly focus on GNNs, a majority of the ML-based approaches to combinatorial problems, and telecommunications problems as well, are GNNs. Also, Zhu et al. [6] conducted a brief review of the existing literature on deep graph generation from various emerging methods to its wide application areas. He et al. [7] provided a review of GNN focused on wireless networks. This work provides a comprehensive survey of the literature of GNNs and their application in wireless networks such as resource allocation, channel estimation, traffic prediction, and so on. Another closely related work, Jiang [4], proposed, for the first time, a comprehensive survey of the huge body of research leveraging various graph-based deep learning models for solving problems in different types of communication networks including wireless, wired, and software-defined networks. In the same direction, Suárez-Varela et al. [8] presented a brief overview of GNN for communication networks including wired and wireless networks. Furthermore, for technical issues in designing and implementing specific GNNs, various works surveyed them from the perspectives of explainability [9], dynamics [10], and hardware-software accelerators [11]. In addition, there are several overviews on applying GNN to solve problems in the IoT [12] and traffic forecasting in urban rail transit systems [13].

On the other hand, several valuable contributions have been made in recent literature on DTs for networking systems. For instance, Nguyen et al. [14] presented briefly how digital twins could be a powerful tool to fulfill the potential of 5G networks and beyond. Wu et al. [15] conducted a survey on DT networks, which exploit DT technology to simulate and predict network dynamics, as well as enhance and optimize network management. The survey provides a comprehensive review of DT networks, covering key features, technical challenges, and potential applications in detail. In the comprehensive survey by Kuruvatti et al. [3], the main contributions can be divided into two parts. In the first part, the survey focuses on reviewing potential use cases and scenarios in a 6G communication network where a DT could play an essential role. The second part focuses on reviewing the activities of the Standards Development Organization on NDTs including a description of key elements, requirements, and reference architecture. Khan et al. [16] presented a taxonomy including twins for wireless and vice-versa. Moreover, Almasan et al. [1] introduced the general architecture of the NDT and provided some potential use cases. Two perspectives of NDTs were briefly discussed including leveraging GNNs for the core components of NDTs and coupling a network optimizer with the NDT. Quantitative results for routing optimization are provided as examples of NDTs. Furthermore, Suhail et al. [17] conducted an extensive review of the state-of-the-art research results on blockchain-based DT, highlighting their key benefits.

However, to the best of our knowledge, no survey has thoroughly investigated the GNN-based DTs for networks. This gap in the existing literature highlights the need for further exploration and research in this area. We summarize the comparison between related works and our paper in Table 1.

Table 1. Summary of related reviews on DT and GNN related topic (✓, x, and * indicate that the topic is totally, not, or partially covered, respectively).

Ref.	Contributions	GNN	DT	Networking Applications
[5]	A comprehensive review of various combinatorial optimization problems on graphs, with a particular emphasis on their applications in the telecommunications domain.	*	x	*
[7]	A comprehensive overview of the application of GNN in wireless networks.	✓	x	*
[4,8]	A comprehensive review of graph-based deep learning models for solving problems in different types of communication networks including wireless, wired, and software-defined networks.	✓	x	✓
[12]	A comprehensive review of recent advances in the application of GNN to the IoT field, including a deep dive analysis of GNN design in various IoT sensing environments.	✓	x	*
[13]	A survey on traffic forecasting in the intelligent transportation system.	✓	x	x
[14]	A brief review on the application of DT with 5G networks and beyond.	x	✓	✓
[15]	A comprehensive survey on the digital twin network including the key features, technical challenges, and potential applications.	x	✓	*
[3]	A review of the potential use cases and scenarios in a 6G communication network where a digital twin could play an essential role.	x	*	✓
[16]	A comprehensive survey on the benefits of twins for wireless and wireless for twins.	x	✓	✓
[1]	Presentation of the application of GNN for the core components of network digital twins and coupling a network optimizer with the network digital twins.	*	✓	✓
Ours	A review of DTs, GNNs, and GraphNDTs in innovating the communication networks.	✓	✓	✓

1.2. Our Contributions

Given the identified gap in the existing literature, our key contributions are outlined as follows:

- A review of DTs applications and use cases for future networks, including both access and core networks.
- A review of several GNN-based models for network management, classified by problems and network domains.
- An analysis of potential applications of DTs based on GNNs for the Next-generation networks (GraphNDTs) and a study of existing works in this direction.
- Insights into challenges and future directions to be taken in order to improve the GraphNDTs to handle large-scale networks and their inherent sub-challenges.

1.3. Structure of the Survey

The rest of our survey is organized as described in Figure 1. Section 2 provides an overview of DTs in the context of future networks, while Section 3 delves into the applications of GNNs for future networks. In Section 4, we analyze the fusion between GNNs and DTs, which we refer to as a Graph neural network-based network digital twin (GraphNDT), with a focus on their potential use cases. We specifically investigate two prominent scenarios: routing optimization and network slicing management. Finally, Section 5 concludes the review by highlighting promising directions and challenges in this evolving field.

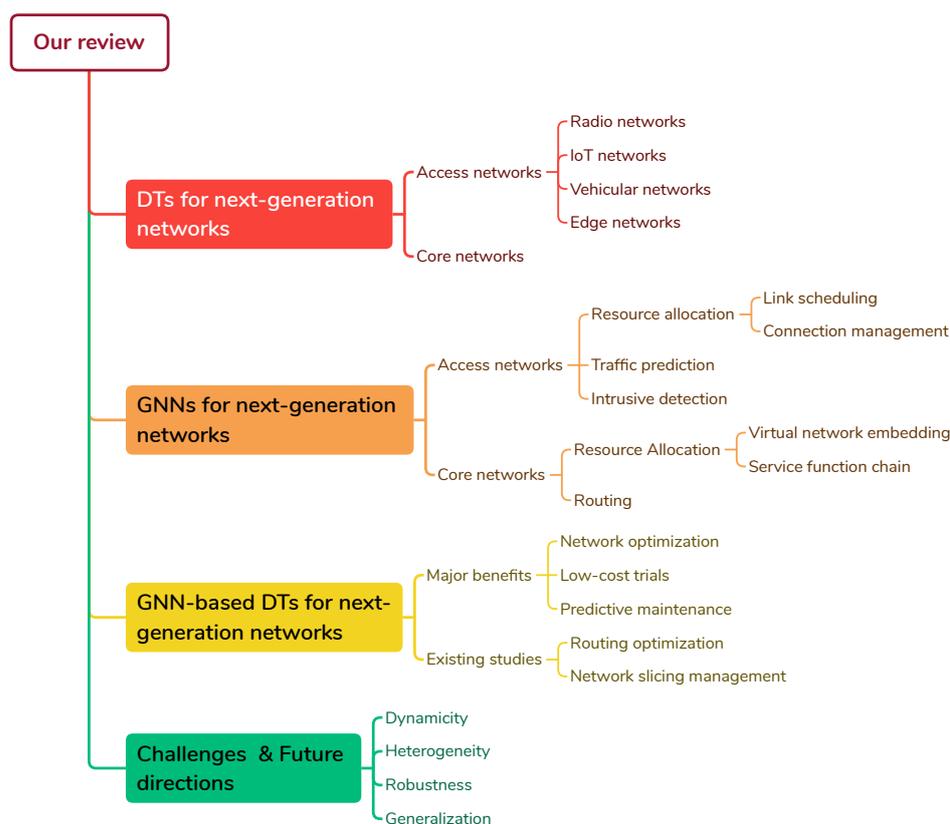


Figure 1. The structure of our paper.

2. Digital Twins for Next-Generation Networks

Digital twins (DTs) emerged in the early 2000s when Michael Grieves introduced them during a course presentation on product lifecycle management. By 2011, implementing DTs was regarded as a challenging process that demanded advancements in various technologies. Although the term “digital twin” emerged in 2003, NASA provided the first detailed account of its application in Technology Roadmaps several years later. In this context, a twin was employed to replicate space conditions and conduct tests for flight preparation. Initially rooted in the aerospace sector, the adoption of DTs expanded to the manufacturing industry around 2012.

Other than the three main components, the digital (virtual part), the real physical product, and the connection between them, according to Tao et al. [18], a DT can be extended to five components by including data and service as a part of it. While sharing a similar purpose with simulations, a digital twin is expected to be more potent by adopting a data-driven approach to model physical objects. Furthermore, the bidirectional synchronization in the connection allows for the real-time updating of the digital replica in response to changes in the mirrored physical object and facilitates the monitoring of the associated product through a virtualization layer. Simulations, on the other hand, focus on modeling the physical entity in specific scenarios or time frames. At the simplest level, a DT is a digital counterpart of a single, atomic, physical thing/system. In a web-based context, at this stage, the DT is a mere proxy of an IoT device with eventually augmented capabilities. However, the digital twin concept can be applied to larger systems in a cascading manner [19], where it can be considered as Things in the Web Of Things vision: they are components of a larger graph of Things and can be composed in a bottom-up or top-down fashion to realize large-scale cyber-physical systems in different application domains, e.g., transportation networks [20–22], water distribution networks [23], smart manufacturing systems [24], etc. In this context, Graph neural networks (GNNs) can be a relevant tool that can help in building additional AI-based services on top of this graph of DTs.

The application of the digital twin paradigm to network management and operation fits into the continuation of global network digitization. It is seen as a high-value target by network operators, and thus there are parallel efforts to offer a standardized definition of what is a Network DT, i.e., DTN or NDT. One can be found in the International Telecommunication Union (ITU) recommendation document [25]: “A digital twin network (DTN) is a virtual representation of the physical network. DTN is useful for analyzing, diagnosing, emulating, and controlling the physical network based on data, model, and interface, to achieve a real-time interactive mapping between physical networks and virtual twin networks. According to the definition, DTN contains four key characteristics: data, mapping, model, and interface [...]”. ITU also proposes a reference architecture of a Network digital twin (NDT) (see Figure 2), including three layers, three domains, and a double closed-loop. The three layers consist of the physical network layer, the Network digital twin (NDT) layer, and the network application layer. Inside the NDT layer, three domains are defined corresponding to three key subsystems: unified data repository, unified data models, and DT entity management. While the unified data repository serves as a single source of data for this layer, and provides the capability to collect, store, serve, and manage data, the unified data model is the ability source of this layer, equipped with specific model instances for different network applications. Within the model domain, an inner closed-loop optimization and emulation is defined between two model types: (i) basic models, which help verify and emulate control changes and optimization solutions before the new configuration is sent to the physical network, and (ii) functional models, which are established for specific use cases and help optimize network configurations to gain better performance. Additionally, the DT entity management enforces the NDT layer with three key controllers: model management, security management, and topology management. Finally, the outer closed loop control feedback is defined based on the three-layer architecture.

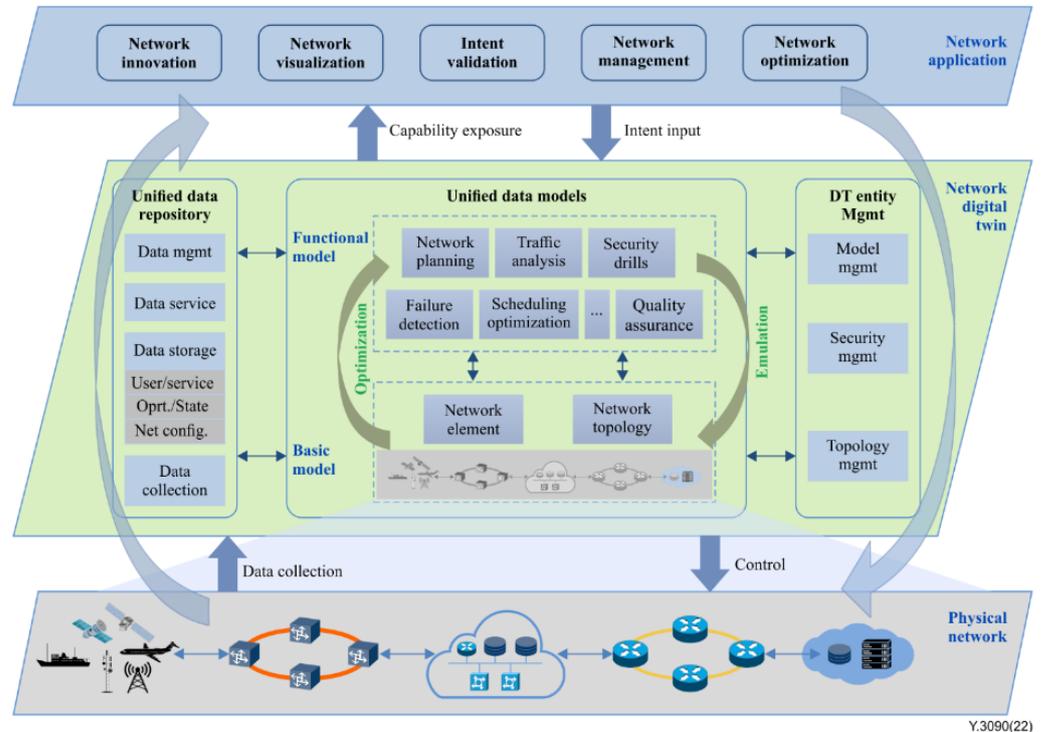


Figure 2. A reference architecture of a Network digital twin (NDT) [25].

Like in many other fields (e.g., IoT, Industry 4.0, and healthcare), the DT paradigm can be applied to a diverse set of use cases. The sections hereafter detail the most notable use cases and associated related work for the NDT.

2.1. Digital Twins for Access Networks

In this section, we provide a survey of existing use cases of DTs in next-generation access networks, spanning across four principle subdomains: radio, Internet of things (IoT), vehicular, and edge networks.

2.1.1. Radio Networks

In 6G THz networks, Line-Of-Sight signal establishment is required to enable high bandwidth and low latency; however, obstacles in the path may absorb THz signals, which makes Line-Of-Sight signal establishment impossible. Pengnoo et al. [26] tackle the problem of signal reflection to avoid obstacles for the signals between a Base station (BS) and a user in indoor amenities, using a digital twin to model and control the signal propagation. The DT uses a set of cameras to stream images of the indoor space. The described system includes modules to perform calculations, e.g., ray tracing, path loss prediction, reflector, and mobile endpoint alignment, allowing the DT estimation of the THz Potential Field (THzPF) and use of this estimation to redirect the signal in real time. The authors present simulation results to show the effectiveness of the system.

In a similar context for outdoor applications, Jiang and Alkhateeb [27] address the adjustment of narrow beams in large-scale MIMO systems. This traditionally requires a large amount of data acquisition/beam sweeping, which scales with the system size/number of antennas. The authors propose to (i) construct a 3D digital replica, i.e., a digital twin, of the real-world communication system, and (ii) use ray tracing algorithms to simulate the signal's propagation. The authors demonstrate that the DT can be used to pre-train Machine learning (ML) models, thus reducing the data acquisition overhead as an important part of the data collection can be simulated. The expected mismatches between the real world and the DT *"can be calibrated by a small amount of real-world data"*.

2.1.2. Internet of Things Networks

Akbarian et al. [28] develop a DT for industrial control systems with the support of an intrusion detection algorithm allowing the detection of attacks and the diagnosis of the types of attack by classification. There exists literature regarding the implementation of DTs for intrusion detection systems but with limitations: one [29] considers only the rule-based detection algorithm, and one [30] shows limitations in the data synchronization between the physical system and its DT. Accordingly, in this work, the authors emphasize the novelty of the detection algorithm and the synchronization ability of the DT enabling continuous synchronization without a need for a specification of the system's correct behavior.

In another study, Benedictis et al. [31] introduce a self-adaptive architecture for DTs aimed at Industrial internet of things (IIoT) anomaly detection. This architecture incorporates ML algorithms and draws inspiration from the MAPE-K [32] (Monitor-Analyze-Plan-Execute over a shared Knowledge base) feedback loop. The authors also demonstrate a Proof Of Concept (PoC) by developing a digital twin for a real-world IIoT system, specifically the European Railway Traffic Management System. The PoC showcases the reference architecture of the DT and includes quantitative evaluation to assess its performance.

2.1.3. Vehicular Networks

In [33], Hui et al. propose an architecture to solve (generic) Federated learning (FL) tasks in heterogeneous vehicular networks (HetVNs). In their simulation scenario, roadside units (RSUs) (i.e., cellular BSs or aerial vehicles) act as FL endpoints with FL capabilities/training resources, while vehicles hold data that can be used in training processes. In their approach, DTs of vehicles and RSUs are deployed, and the FL multi-tasks are considered as a matching game between learning task requests and available data within the RSUs range, which must be optimized in terms of training cost and model accuracy.

Similarly, Zhao et al. [34] deal with optimizing software-defined vehicular networks where the DT acts as a centralized controller of the network, enabling more computation resources than what is available at the edge. In this centralized configuration, the DT con-

troller provides optimal per-flow routing computation adapted to the demands of vehicles. The authors implement a simulation where the vehicular network state is considered as a temporal graph, constructed as a Hidden Markov Model, and the routing scheme optimization is a temporal graph routing task. The centralized DT can run different routing scheme simulations by prioritizing different parameters, e.g., historical routing or vehicle density, and then apply the best routing scheme found to improve routing efficiency, maintenance, traffic flow, and security in the physical network.

2.1.4. Edge Networks

To achieve 6G ambitions such as ubiquitous connectivity, extremely low latency, and enhanced edge intelligence, Multiple-access edge computing (MEC) plays a crucial role. Lu et al. [35] propose a wireless digital twin edge network model that aims to provide hyper-connected user experience and low-latency edge computing. They tackle the *DT to edge* association problem concerning the dynamic network states and varying network topology, where DTs can be either twins of user devices or twins of services that users are using. The authors propose to use Deep reinforcement learning (DRL) for digital twin placement and transfer learning for digital twin migration to follow dynamically placed users, thus trying to minimize the latency and energy costs of users in the network.

In a different context, Van Huynh et al. [36] focus on the integration of MEC offloading with Ultra-reliable low-latency communication (URLLC) and short packet transmission, specifically in the IIoT context. The authors aim to minimize the latency of task offloading by optimizing the user association, transmission power, and processing rate of User equipments (UEs) and edge servers (ESs). To achieve this, a DT of the edge network architecture is constructed, while the wireless communications between UEs and ESs are established via URLLC links, realizing a *DT-empowered URLLC edge network*. In this work, the DT replicates the physical system (hardware information, operating applications, real-time states), optimizes resources, and makes decisions to control the whole system in real time. To achieve this, the DT optimizes a variety of variables, e.g., user association, offloading policies, transmission power, processing rates, energy consumption budget, and computation resources budget of UEs and ESs. Similarly, Do-Duy et al. [37] introduce the use of DTs for intelligently offloading the computing tasks of UEs onto MEC servers. They formulate the problem as an optimization problem with the main objective of minimizing the total digital twin latency by choosing the optimal user association, transmit power, offloading policies, and processing rate of UE and MEC servers.

Furthermore, Duong et al. [38] presented a solution to address the challenge of minimizing latency in the context of intelligent offloading of IoT devices, assisted by digital twins, onto edge networks deployed with unmanned aerial vehicles (UAVs). The solution takes into consideration the constraints imposed by URLLC links. The problem is formulated as an optimization task that involves jointly optimizing the transmit power, offloading policies, and processing rate of IoT devices and ESs.

2.2. Digital Twins for Core Networks

As for the core networks, DT also plays an important role in their evolution to the Next-generation network. As an example, Wang et al. [39] propose a DT framework to enhance optical communication networks. The authors describe a generic framework composed of the physical layer, the data layer, the model layer, and the application layer. The approach includes the propositions of three separate Deep learning (DL) models to achieve different objectives. First, a fault management model that uses bidirectional Gated recurrent unit (GRU) [40] to predict faulty equipment and XGBoost [41] to make a fault diagnosis. Second, a flexible hardware configuration model to dynamically configure a programmable optical transceiver (POT) using DRL. Finally, a dynamic transmission simulation system based on bidirectional Long short-term memory (LSTM) [42] that could replace a traditional block-based optical transmission simulation system.

Another approach proposed by Seilov et al. [43] involves building networks of digital twins to tackle problems in complex telecommunication networks with two specific examples in feedback loops and traffic monitoring. In the former case, the network of digital twins allows the developer the tracking of all changes made and intervention in the development of the telecommunication system. In the latter case, the network of digital twins enables the network operators to solve urgent problems in emergencies where the network traffic suddenly changes due to the dysfunction of some network elements.

In another context, Yigit et al. [44] present a digital twin-enabled framework aimed at Distributed denial of service (DDoS) attack detection for autonomous core networks. Since existing DDoS solutions are insufficient for data centers and edge networks in terms of scalability, detection rates, and latency, the authors develop an online ML-based algorithm for effective DDoS detection. This algorithm processes the data that can be captured in real time with the support of the digital twin designed for the core network. Moreover, the Yet another next generation (YANG) model and automated feature selection are also involved in reducing the complexity of the provided data before feeding it to the ML algorithm. Finally, the authors evaluate and show the outperformance of their proposed system using two different DDoS attack datasets as the simulation of the core network.

Lessons Learned

DTs have emerged as valuable tools in Next-generation networks, offering advanced capabilities in network optimization, monitoring, and security across various domains, including core networks, access networks, edge networks, and vehicular networks. In the context of optimization, digital twins function as centralized controllers that collect data from the physical network and propose optimal policies. For monitoring, they operate as a closed-loop autonomous system, ensuring effective network surveillance. Additionally, this autonomous system coupling with intrusion detection further emphasizes their importance in ensuring network security. Overall, digital twins play a pivotal role in enhancing network performance and safeguarding network integrity.

3. Graph Neural Networks for Next-Generation Networks

This section provides an introduction to Graph neural networks (GNNs), followed by their applications in the access and core network of the next-generation network.

3.1. Backgrounds on Graph Neural Networks

From now, unless otherwise specified, we denote \mathcal{G} as a graph and $\mathcal{G} = (V, E)$, where V is the set of $|V| = n$ vertices and E is the set of edges (v_i, v_j) for $v_i, v_j \in V$ if there exists an edge between them. The set of edges can also be represented under the form of an adjacency matrix A , which is a binary matrix of size $n \times n$, $A_{ij} \in \{0, 1\}$ where 1 (or 0) means the existence (or the nonexistence) of the edge (v_i, v_j) .

Based on the design patterns of the models, we can classify GNNs into two groups: (i) *Spectral-based GNNs* and (ii) *Message-Passing-based (or spatial-based) GNNs*.

3.1.1. Spectral-Based Graph Convolutional Networks

Convolutional neural networks (CNNs) dominate image processing due to their ability to extract information from local neighborhoods of pixels (i.e., receptive field) in images. This property can also be leveraged in designing GNNs, where nodes can be seen as pixels, and a GNN layer acts like a CNN layer, capturing local information from node neighborhoods. However, defining a translation-invariant operator in the vertex domain is challenging. Graph signal processing (GSP) theory offers a solid foundation to formulate convolution operators in the spectral domain [45,46].

In GSP, a Graph Fourier transformation (GFT) is defined for an undirected graph with a symmetric adjacency matrix A . The graph signal $x \in \mathbb{R}^n$ represents the scalar signals of each node in V [45]. The graph Laplacian matrix $L = D - A$ is a fundamental operator in GSP, where $D \in \mathbb{R}^{n \times n}$ is the diagonal matrix such that $D_{ii} = \sum_{j=1}^n A_{ij}$, i.e., the i th node

degree. Based on the GFT, Defferrard et al. [46] propose the K -order approximation of the convolution, using the Chebyshev expansion to reduce the learning and computational complexity. The graph convolution is approximated as

$$g_{\theta}(x) \approx \sum_{k=0}^K \theta_k T_k(\tilde{L})x, \tag{1}$$

where, via the recurrence of the Chebyshev polynomials, we compute $T_k(\tilde{L}) = 2\tilde{L}T_{k-1}(\tilde{L}) - T_{k-2}(\tilde{L})$ and $\tilde{L} = 2L/\lambda_{\max} - I_n$ given that λ_{\max} is the largest eigenvalue of the Laplacian, $T_0 = 0$, and $T_1 = \tilde{L}$. Parameters θ can be updated to optimize the learning objective. Li et al. [47] extend the spectral-based graph convolution for directed graphs by using the same spectral approach. It is also important to note that K can be interpreted as the kernel size of the convolution, i.e., the radius of the local neighborhood from the central node.

3.1.2. Message-Passing-Based Graph Neural Networks

Kipf and Welling [48] consider convolving the graph signal x with a kernel of size $K = 1$. Additionally, they reduce the number of learned parameters to avoid overfitting. Accordingly, the convolutional operator is approximated as

$$g_{\theta}(x) \approx \theta \left(I_n + D^{-1/2}AD^{-1/2} \right) x. \tag{2}$$

By stacking k first-order convolutional layers together, we can approximate the k -order graph convolution. Although this is often referred to as the Graph convolutional network (GCN), many other GNNs are also designed with the same architectural pattern, which is to stack several sub-modules together. Gilmer et al. [49] propose the generic framework Message-passing neural network (MPNN) to characterize this architectural pattern where these sub-modules could be described in one function:

$$h_u^{(t)} = \text{Update}^{(t)} \left(h_u^{(t-1)}, \text{Agg}^{(t)} \left(\left\{ h_v^{(t-1)}, \forall v \in \mathcal{N}(u) \right\} \right) \right) \tag{3}$$

$$= \text{Update}^{(t)} \left(h_u^{(t-1)}, m_u^{(t-1)} \right), \tag{4}$$

where $h_u^{(t)}$ is the *hidden* embedding of node $u \in V$ before being passed to layer t , and $h_u^{(0)} = x_u$ is the initial node features. In other words, an MPNN layer defines two important *differentiable* functions (neural network as an example): *Update* and *Agg* (-regation). After the aggregation of the neighboring node embedding of node u into *message* $m_u^{(t-1)}$, the hidden embedding of node u is updated using its previous node embedding and the aggregated message. A simple example of an MPNN is illustrated in Figure 3. The GCN can also be regarded as an MPNN with the addition as *Agg* and the learnable weighted sum as *Update*. In the same vein, we can consider the Graph attention network (GAT) [50] as an MPNN as well with the multi-head attention mechanism as *Agg* and the concatenation as *Update*. We describe the updated message on node u after the t th layer of deep one-head attention GAT using equations below:

$$\alpha_{u,v} = \frac{\exp(\text{LeakyReLU}(a^\top [Wh_u || Wh_v]))}{\sum_{v' \in \mathcal{N}(u)} \exp(\text{LeakyReLU}(a^\top [Wh_u || Wh_{v'}]))}, \tag{5}$$

$$m_u = \sum_{v \in \mathcal{N}(u)} \alpha_{u,v} h_v, \tag{6}$$

$$h_u^{(t)} = [h_u^{(t-1)} || m_u^{(t-1)}], \tag{7}$$

where $||$ is the concatenation operator, LeakyReLU [51] is the non-linear activation function, a and W are learnable weights. For the sake of clarity, we omit superscript $(t - 1)$ from

$\alpha_{u,v}$, a , W , h and m in Equations (5) and (6). The extension to multi-head attention [52] is straightforward.

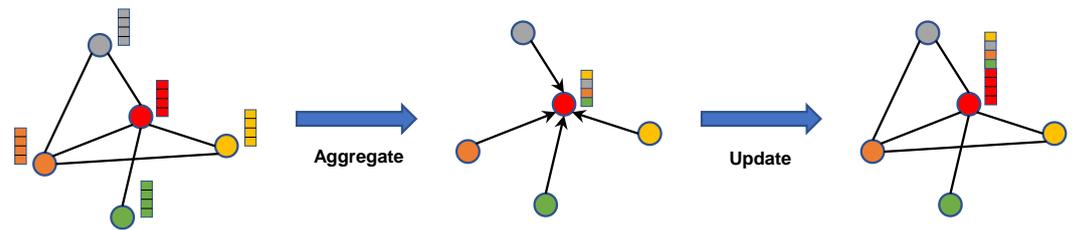


Figure 3. A simplified illustration of a Message-passing neural network (MPNN) on an undirected graph where we aggregate neighbor features by “mixing” them and updating the target node (the red one) using concatenation.

Overall, the main difference between spectral-based GCNs and MPNNs is their design patterns. While spectral-based GCNs provide a rigorous formulation for graph convolution, MPNNs simplify the computation, enable modularization, and facilitate solution design. In the following discussion, all GNNs are designed based on these two frameworks. Without further clarification, the term GCN refers to spatial-based GCN. Until now, GNNs continue to be a highly active research direction which still presents several challenges in both theoretical and technical aspects. However, in this section, and throughout this survey, our focus is exclusively on their applications in concrete networking problems and within the context of the Digital twin (DT). For a more in-depth exploration of challenges and current progress in this research field, we recommend readers to refer to [53–56].

3.2. GNN in Access Networks

In this section, we examine the current body of literature that explores the application of graph-based approaches in the management of access networks. Our review specifically centers around three key areas: *resource allocation*, *traffic prediction*, and *network security*, particularly *intrusion detection*.

3.2.1. Resource Allocation

Resource allocation in access networks that we focus on in this review mainly concerns *connection management* and *link scheduling*, as summarized in Table 2.

Table 2. Summary of application of GNN-based models for resource allocation in access networks.

Ref.	Model	Baseline	Objective	Metric
[57]	DRL, GNN	Default	Throughput and coverage maximization and load balancing	Gain percentages
[58]	DRL, GCN	Random, CNN+DRL, PG	Data rate maximization	Data rate and convergence performances
[59]	DQN, GAT, A2C	DQN, A2C, Hard slicing	Data rate maximization while guaranteeing QoS	Utility performances
[60]	GCN, Spectral clustering	Max. achievable and max. sum of rate and power	Power allocation and user association	Data rate
[61]	GCN, DNN	Default	Sum rate maximization	Accuracy, average sum rate
[62]	GCN, MWIS	Local greedy solver	Delay minimization	Accuracy, average sum rate

Connection Management

Playing a crucial role in resource management within access networks, connection management aims to achieve smooth, balanced, and fair throughput. Conventional approaches

to connection management, particularly user-cell association, often rely on sub-optimal and greedy solutions, such as connecting each user to the cell with the highest received signal strength. However, the performance of networks can be enhanced by harnessing the potential of Machine learning (ML)-based solutions.

The advent of the next-generation software-defined 5G networks, as defined by the Open Radio Access Network (O-RAN) alliance, facilitates the integration of ML/AI-based techniques to address various challenges in access networks. In this context, Orhan et al. [57] propose a novel approach to handle user association and optimize load balancing on top of the O-RAN network architecture using GNNs. The RAN infrastructure and its connections to User equipments (UEs) are conceptualized as a graph of nodes representing radio cells and UEs, and edges encompassing (i) connections between adjacent cells and (ii) association between each UE and its associated cell. GNNs are then proposed as a data-driven method to comprehend and extract features from the underlying graph structure to the end of optimizing connection management based on three quantitative metrics. Accordingly, the optimization involves the joint maximization of the sum of UE throughput, coverage, and the load balancing index.

Similarly, Zhao et al. [58] leverage GNNs to address the resource management in the Radio access network (RAN). Their focus is primarily on the cognitive radio context, where unlicensed secondary users (SUs) can occasionally access the idle resources of licensed primary users (PUs). In line with the previously introduced work [57], they represent the underlay cognitive radio network as a graph, with vertices representing UE-cell links and edges denoting the interference between them. The radio resource management is then formulated as a decision-making problem where an actor proposes jointly the channel and power allocation while ensuring the constrained proportion of resources between PUs and SUs to maximize the data rate for all users. To achieve this, the authors employ GNNs to exploit the interference information embedded in the constructed graph for the generation of allocation policies. Additionally, they consider the modeling of the users' mobility, resulting in a dynamic graph with a continuous decision-making mechanism. Consequently, the authors adopt a Deep reinforcement learning (DRL) framework for model learning. The simulation results demonstrate the feasibility and convergence of the proposed scheme, showing a significant performance improvement compared to existing approaches.

In the context of network slicing where diverse services are provided over the same communication infrastructure, resource allocation becomes challenging, especially in dense cellular network scenarios with multiple slices and Base stations (BSs). The key difficulty lies in designing a real-time inter-slice resource management strategy that can handle frequent BS handovers and accommodate varying service requirements. To address this challenge, Shao et al. [59] propose a Multi-agent reinforcement learning (MARL) [63] solution, where each agent represents a BS. The authors utilize a GAT to enhance the temporal and spatial cooperation between agents and develop a data-driven inter-slice resource allocation strategy in real time. The effectiveness of GAT in improving DRL in multi-agent systems is demonstrated through experiments. GAT is employed in conjunction with both the value-based technique *Deep Q-network* (DQN) [64] and a hybrid approach combining policy-based and value-based methods *Advantage actor-critic* (A2C) [65], showcasing its potential in enhancing performance and efficiency.

Furthermore, Hou et al. [60] aim to optimize radio resource allocation and user association within the context of the Ultra-Dense Network (UDN), where BSs including both micro- and macro-cells are deployed densely. The optimization process is centered on leveraging Channel state information (CSI) and is structured as a synchronous two-stage procedure: user association and power allocation, with the ultimate goal of maximizing the total data rate. To associate UEs to the set of micro- and macro-cells, they propose constructing an undirected graph with nodes as UEs, and edges are weighted based on the similarity between UEs features, which are composed of downlink channel gains from an UE to all the BSs. The spectral clustering algorithm is then employed on this graph to cluster UEs and associate them with appropriate cells. Simultaneously, a GNN model

with learnable parameters is employed to capture the interference information and allocate radio power. A graph of BSs is utilized, with node features comprising channel gains of each BS to all users, and unweighted edges connecting cells with overlapping coverage. The GNN updates node features with captured interference information, feeding them into a fully connected network to generate power allocation strategies for every pair of UE and BS. The simulation shows that the method yields superior results and is less time consuming than the existing algorithms within the context of UDN.

Link Scheduling

In access networks, link scheduling involves allocating and managing limited bandwidth resources among multiple users or devices. The goal is to ensure fair and optimal distribution of bandwidth, taking into account individual requirements and the network's capacity. Efficient link scheduling algorithms aim to minimize congestion, reduce latency, maximize throughput, and maintain fairness among users.

In general, link scheduling for device-to-device (D2D) communications is formulated as an NP-hard non-convex combinatorial problem. Traditional methods predominantly rely on mathematical optimization techniques, often requiring accurate CSI, which can be resource-intensive to obtain. A recent alternative involves the application of GNNs to address this issue. In [61], a graph embedding-based method is introduced to achieve link scheduling in D2D communications, eliminating the need for CSI. This method consists of constructing a fully connected directed graph for the D2D network where each D2D pair represents a node and interference links are edges. Input node features and interference links are computed based on spatial distances between devices in each D2D pair and distances between each pair, respectively. This is followed by a deep learning architecture to capture structure-aware node embeddings. Finally, the link scheduling problem is formulated into a binary classification problem to determine whether a D2D link should be deactivated. Therefore, given extracted node embeddings, a multi-layer classifier can generate a link scheduling strategy. Extensive simulation demonstrates that the proposed method performs near-optimally compared to state-of-the-art methods and requires only hundreds of training network layouts. Furthermore, it proves to be competitive in terms of scalability and generalizability to more complex scenarios.

In wireless multi-hop networks, delay holds significant importance for various applications. However, existing max-weight scheduling algorithms focus on instantaneous optimality and may not perform well in delay-oriented scheduling scenarios. To address this issue, Zhao et al. [62] propose a delay-oriented distributed scheduler based on GCNs. The proposed scheduler employs a GCN model to generate node embeddings, capturing both the network topology and multi-step lookahead backlogs. By considering the relationship between current backlogs and the schedule of the previous time slot, the scheduler can make more informed scheduling decisions. In wireless networks that are small to medium-sized and exhibit heterogeneous transmit power, particularly with central links having numerous interfering neighbors, the proposed distributed scheduler surpasses myopic schedulers that rely on greedy and instantaneously optimal maximum weighted independent set solvers. The solution showcases strong generalizability across various graph models while introducing minimal communication complexity overhead.

3.2.2. Traffic Prediction

Traffic prediction aims to anticipate the volume of network traffic based on historical data. This proactive measure helps to prevent future congestion and allows for dynamic optimization of network resources [66]. Like with IDS, traffic prediction can greatly benefit from GNN-based models due to their ability to comprehend and model network data effectively. In other words, as the traffic at a particular node depends not only on its historical values but also the traffic conditions of its near or far neighboring nodes, GNN combined with some temporal models (e.g., Recurrent neural network (RNN) or Long short-term memory (LSTM) model) can well model these dependencies [67].

In this context, Wang et al. [68] use GNN with RNN to extract spatiotemporal dependency from inter-tower and in-tower traffic and predict cellular traffic at a large scale. The proposed model outperforms the state-of-the-art approaches by 13.2% and 17.5% in terms of MAE and MAPE, respectively. He et al. [69] introduce the *Graph Attention Spatial-Temporal Network* (GASTN) to forecast the mobile traffic. GASTN takes into account spatial dependencies by utilizing a geographical relation graph. Moreover, the model integrates RNN to extract temporal features within sequential data. Additionally, GASTN introduces two attention mechanisms to incorporate two distinct effects holistically. By incorporating the attention mechanism, GASTN can consider various factors and their importance when predicting traffic. In another work, Yang et al. [70] present *STEM-GCN*, a new GCN exploiting semi-variogram, which is a measure of dissimilarity between two variables across the domain of spatial distance and time lags. This model effectively captures spatial and temporal dependencies from dynamic graphs, includes a correlation smoothing strategy to reduce noise and improve link prediction accuracy, and handles network dynamics by propagating spatial and temporal characteristics by using stacked memory cell structures across sequential time steps.

Kalander et al. [71] propose the *Spatio-Temporal Hybrid Graph Convolutional Network model* (STHGCN), which uses Gated recurrent units (GRUs) to model temporal dependencies and a hybrid GCN to capture complex spatial dependencies from three perspectives: (i) spatial proximity (SP), (ii) functional similarity (FS), and (iii) recent trend similarity (RTS). As shown in Figure 4, the model processes temporal slices of features from each BS and applies the hybrid graph CNN using the three aforementioned spatial relationships. The output is processed by a GRU along with external information such as weather data or metadata and is then handled by a fully connected network to generate the final traffic prediction.

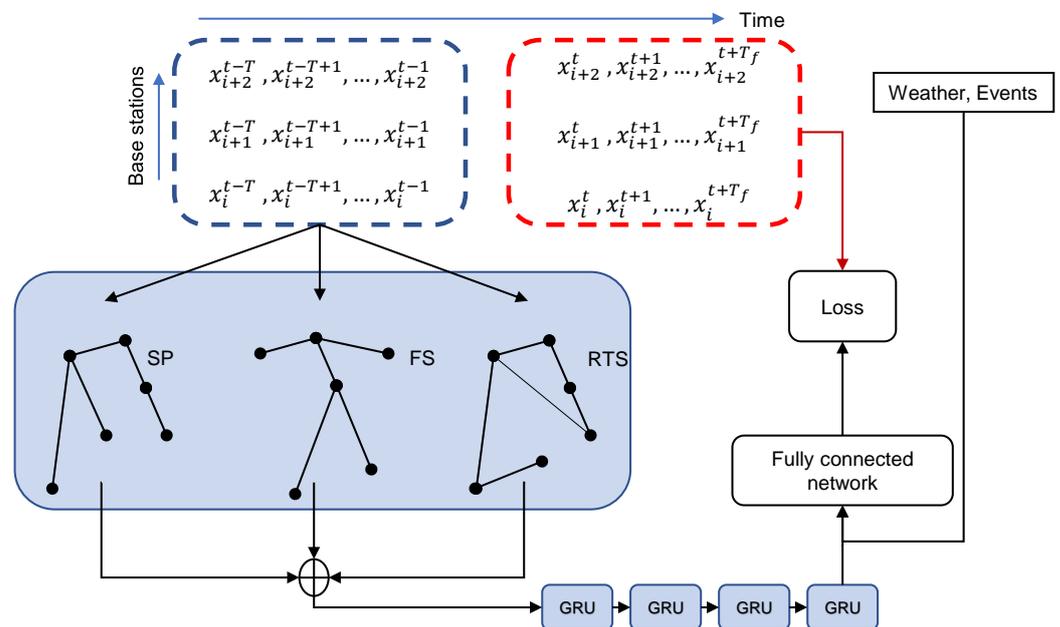


Figure 4. The architecture of STHGCN [71].

While the aforementioned approaches have demonstrated efficiency in traffic prediction, they do not take into account the similarity among different types of cellular services (e.g., calls, internet) and regions. To address this issue, Zhou et al. [72] introduce a transfer learning strategy based on a GCN model for large-scale traffic prediction called *STA-GCN*. The proposed model integrates transfer learning, an attention mechanism, the CNN model, and GCN to capture both temporal and spatial dependencies. The experiments conducted with the Telecom Italia dataset validated the efficacy of transfer learning in enhancing knowledge reusability and accelerating model fitting, without com-

promising prediction accuracy. Moreover, traffic in a wireless network is impacted not only by historical traffic and cross-domain datasets but also by handover traffic from the BS. Consequently, Zhao et al. [73] propose the *STGCN-HO* model, which uses a handover graph for cellular traffic prediction. The model leverages the transition probability matrix of the handover graph to enhance network traffic forecasting and fuses features from auxiliary data (e.g., day of the week, hour of the day), spatial, and temporal domains. Evaluations performed with real-world 4G LTE traffic data show that *STGCN-HO* outperforms the baseline models, including Historical average (HA) [74], Auto-Regressive Integrated Moving Average (ARIMA) [75], and LSTM models [76], at both the cellular and BS levels. However, all those models primarily focus on predicting future traffic load for a city, an urban area, or a BS, which is vague for fine-granular user-level traffic prediction. In this context, a fine-grained prediction is proposed by Yu et al. [77], where the authors present a spatio-temporal fine-granular user traffic prediction mechanism for cellular networks called *STEP*. Specifically, *STEP* is based on the integration of GCN and GRU models to capture the spatiotemporal features of the individual user traffic. In addition, Wang et al. [78] introduce an attention mechanism to assign appropriate weights to each node. They propose a time-series similarity-based graph attention network for cellular traffic prediction known as *TSGAN*. This model uses dynamic time warping [79] to compute the time-series similarity between the network traffic of every pair of cells and GATs to extract spatial features. Comparative experiments conducted on the Telecom Italia and Abilene datasets against GNNs and GRU models demonstrated the superior performance of *TSGAN*.

To conclude, we summarize the reviewed methods along with their experiment set-ups and quantitative results in Table 3.

Table 3. Summary of application of GNN-based models for traffic prediction.

Ref.	Models	Dataset	Baselines	Prediction Horizon	Performance
[68]	GNN	Unpublished	ARIMA, HW, LSTM	30 min	MARE: 0.79
[69]	GNN, RNN, Attention mechanism	Telecom Italia	HA, ARIMA, MLP, LSTM, CNN-LSTM	10 min	MAE: 30.93
[71]	GCN, GRU	Unpublished	HA, GCN, Attention GCN, DCRNN, Graph WaveNet	15 min	MAE: 21.7, RMSE: 47.2
[72]	Transfer learning, Attention mechanism, CNNs, and GCN	Telecom Italia	LSRM, STGCN, ASTGCN, DCRNN	1 h	MAE: 55.46, RMSE: 116.92
[73]	GCN, Gated Linear Unit	Unpublished	HW, ARIMA, LSTM	15 min	RMSE: 3.91×10^5
[77]	GCN, GRU	Unpublished	ARIMA, LSTM, GNN	5 s	MARE: 2.96×10^4
[78]	GNN, GAT	Telecom Italia	GNN, GRU	10 min	MAE: 25.75, MAPE: 0.13, RMSE: 35.94

3.2.3. Intrusive Detection

As Industrial IoT (IIoT) networks increasingly permeate diverse sectors like home, industry, healthcare, and agriculture, the application of learning algorithms has become paramount for enhancing communication processes and real-time analytics. Yet, the rapid expansion of IoT devices brings substantial security concerns. The absence of robust security policies and standardization for IIoT networks exposes these systems to a high risk of malicious attacks. In this scenario, GNN emerges as a promising solution for next-generation IoT/IIoT networks. GNN is one of the potential tools for data representation and processing, enabling effective modeling and analysis of complex relationships within access networks. By leveraging GNN capabilities, it becomes possible to bolster real-time analytics and, consequently, improve the management and protection of the network and devices.

Intrusions are characterized by a sequence of suspicious and benign interactions between networks or processes in a host system. Thus, depending on where the detection takes place, Intrusion Detection Systems (IDS) can be divided into two broad categories, which are network-based intrusions (NIDS) and network host-based intrusions (HIDS) [80]. NIDS consistently monitor and analyze network traffic to detect network attacks. They can effectively detect malicious activities and alert network administrators to take action and mitigate security threats. In this subsection, we focus on the application of GNN to both NIDS and HIDS. However, given our interest in the application of GNN within Next-generation networks, our primary emphasis is on GNN-based strategies for NIDS, as briefly summarized in Table 4.

Table 4. Summary of application of GNN-based models for IDS.

Ref.	Learning	Target	Models	Dataset	Performance
[81]	Supervised	Node	GCN	-	Acc: 99.51%, 99.03%
[82]	Supervised	Node	Inferential SIR-GN	CAIDA + synthetic data	F1: 97.85–99.78%
[83]	Supervised	Node	GNN, GRU	CIC-IDS2017	F1: 99%
[84]	Supervised	Node	Graph network	CTU-13 + synthetic data	ACC: 96%
[85]	Supervised	Edge	E-GraphSAGE, GraphSAGE	BoT-IoT, ToN-IoT, NF-BoT-IoT, NF-ToN-IoT	F1: 100%, 99%, 97%, 100%
[86]	Supervised	Node	E-GraphSAGE, GAT	UNSW-NB15, CIC-Darknet2020, ToN-IoT, CSE-CIC-IDS2018	F1: 99.5%, 92.32%, 99.88%, 96.5%
[87]	Supervised	Node	GDN, GAT	SWaT, WADI	Acc: 99%, 98%
[88]	Semi-supervised	Node	GCN	CTU-13, Honeypot dataset	F1: 98.27%, 98.22%
[89]	Semi-supervised	Edge	GNN, Autoencoder, Attention mechanism	LANL2015, CERT, PicoDomain	F1: 89.28%, 91.28%, 92.68%
[90]	Supervised	Node	GIN, GNNExplainer	-	F1: 99.52%, 99.47%
[91]	Supervised	Node	DB-GNNExplainer	-	F1: 99.14%

As depicted in Figure 5, conventional ML/Deep learning (DL) models typically rely on statistical features from each network flow for training, without considering the topological structure. In contrast, GNNs incorporate topological information along with network flow features, which enables them to model spatial relationships and dependencies between nodes (for example, traffic). This feature becomes particularly beneficial when the network data's structure is complex and non-Euclidean. GNN is capable of handling dynamic topologies where nodes and edges can fluctuate over time. Additionally, GNN can detect subtle patterns in the graph structure data. For instance, GNNs can identify abnormal communication patterns between certain nodes, a capability that outperforms traditional ML/DL models. Zhou et al. [81] propose to detect with an End-to-end (E2E) data-driven approach using the GCN model. The experiment results showed that GCN performs better than the logistic regression model. Addressing the potential issue of overfitting with GNN models, Carpenter et al. [82] introduce an approach termed Inferential SIR-GN. This technique is designed to generalize unseen data within large graphs while prioritizing node structural similarity, thereby enhancing the robustness of GNNs in intrusion detection scenarios. In another GNN-based NIDS approach, Pujol-Perich et al. [83] focus on revealing the relationships between flows using a GNN model. They add new nodes to represent each flow GNN model. Specifically, for each flow, there are three nodes: the source host node, the destination host node, and the flow node, and the proposed graph includes heterogeneous elements (i.e., hosts and flows). Subsequently, a message-passing algorithm is proposed to efficiently learn from the host connection graph. The performance of their model was compared with some advanced ML models using the CIC-IDS2017 dataset, showing superior results in terms of the F1 score. Moreover, Protogerou et al. [84] provide an anomaly detection solution within IoT systems by employing a GNN-based model.

They propose a multi-agent system based on GNN, where each agent is implemented using a GNN that can learn the representation of physical networks. Such an approach helps to explore the collaboration of intelligent agents to efficiently detect the intrusion. Lo et al. [85] also use the GNN for IoT network intrusion detection, called *E-GraphSAGE*. *E-GraphSAGE* enables the collection of information on flow-based features and topological patterns to learn the patterns of network flows and hence to support the detection of malicious edges. Using four benchmark datasets, the simulation results show that the *E-GraphSAGE* performs better than ML-based classifiers such as XGBoost. Similarly, Chang and Branco [86] extend *E-GraphSAGE* by proposing an edge-based residual graph attention network (*E-ResGAT*). *E-ResGAT* uses an attention mechanism supporting edge features and embedding residual learning to enhance malicious traffic detection. The obtained results show that *E-ResGAT* outperforms the original model *E-GraphSAGE*. In the same direction, Deng and Hooi [87] propose a structure learning approach in combination with GNN and use an attention mechanism called *GDN*. The novelty of their approach lies in using a distance metric to discern the relationships between nodes, primarily by selecting the top-K closest ones as the neighbor dependencies. Subsequently, a graph attention convolution is utilized to encapsulate the process of information propagation. Experiments on two real-world sensor datasets show that *GDN* can handle high-dimensional time-series data and outperform baseline models. In addition, Sun and Yang [89] introduce a lateral movement (LM) detection system called *HetGLM*. This system constructs a heterogeneous graph consisting of users, devices, processes, and files. LM detection is redefined as an anomaly link detection task within this heterogeneous graph. The model utilizes meta-paths within the graph to profile each network entity and identify authentication activities that deviate from benign behavior. While a few benign samples are required as input, the model benefits from the use of semi-supervised learning, allowing for more efficient detection of lateral movement.

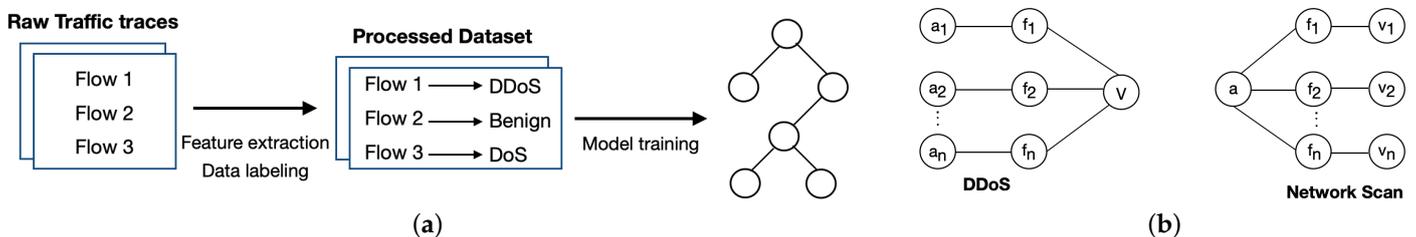


Figure 5. (a) Scheme of Conventional machine learning-based NIDS using network flow. (b) Graph-based representation of some known attacks, a_n nodes refer to the attackers, v_n nodes represent the targets, and f_n nodes represent the different flows.

Despite the performance of the abovementioned works, they are based on the supervised GNN model, where labeling data is often difficult and time-consuming. To handle this issue, semi-supervised learning can be used as an alternative solution. It exploits abundant unlabeled data in combination with a small amount of labeled data [92]. In this, Zhao et al. [88] propose a novel bot detection framework, namely *Bot-AHGCN*. *Bot-AHGCN* models fine-grained network flow objects into a heterogeneous graph and transforms the bot detection problem into a semi-supervised node classification task on the graph. The node is one flow entity from the 6-tuple (IP_src, IP_dst, port, protocol, request, response), and edges are actions between flow entities such as *access* or *connect*.

The black-box nature of GNN models poses a challenge to both the trust and understanding of the final results. This may raise several questions for the operator, such as the following: On what information do the models base their decisions on when determining whether a node is abnormal? Is the information learned by the model comprehensible? To address these concerns, researchers propose explainable GNN models for botnet detection [90,91]. Specifically, Lo et al. [90] develop an automated botnet detector and a forensic tool called *XG-BoT*. To overcome the limitations of the GCN regarding the number of layers,

XG-BoT integrates group-reversible residual connections with the Graph Isomorphism Network (GIN). To enhance comprehensibility further, it offers explanations via GNNExplainer [93] and saliency maps. GNNExplainer underscores the subgraph structure and node-level features pertinent to the final prediction. Using the dataset proposed in [81], the simulation results demonstrate that XG-BoT achieved state-of-the-art performance in terms of the F1-score. Similarly, Zhu et al. [91] propose an interpretive method for GNN-based botnet detection called *BD-GNNExplainer*. This method first employs a GNN model to detect a botnet attack; next, BD-GNNExplainer identifies the most informative edges that are heavily focused on during the training process of the GNN models. Finally, an interpretability score is assigned by comparing the identified edges with the ground truth. The proposed approach aids not only in accurately identifying threats but also in understanding the reasoning behind these identifications. The results show that a model with better classification performance is not necessarily more interpretable.

3.3. GNN in Core Networks

In core networks, the two main topics include resource allocation and routing.

3.3.1. Resource Allocation

In this section, we discuss resource allocation within the core network, with the common problems of Virtual network embedding (VNE) and Service function chain (SFC), as summarized in Table 5.

Table 5. Summary of application of GNN-based models for resource allocation in core networks.

Ref.	Model	Baseline	Objective	Metric
[94]	GCN, DRL	R-ViNE, D-ViNE, GRC, MCVNE, and NodeRank	VNE	Acceptance ratio, average revenue latency, node, and link resource utility
[95]	GNN, DRL	DDQN, MSGAS, Eigendecomposition	VNF placement	SFC rejection ratio, computation time
[96]	GNN, DRL	OFM, size-greedy, pairwise, random	Reduce flow migration cost	Migration cost, computation time
[97]	GNN, Kmeans	FirstFit, BestFit, GRC, NeuroViNE	Improve runtime and performance	Parallelizability, acceptance ratio, revenue and cost, CPU and link utilization
[98]	GNN, RL	NodeRank, MCST-VNE, GCN-VNE	Dynamic VNE, reduce resource fragmentation	Revenue, acceptance rate
[99]	GNN, ILP	DNN-based model	SFC	Cost (delay): average, fail ratio, overmax
[100]	GCN, DRL	LDG, DNN-DDQN	SFC	E2E delay

VNE involves the placement of virtual network services onto substrate network components. The performance of VNE algorithms plays a crucial role in determining the effectiveness and efficiency of a virtualized network, making it a critical component of network virtualization technology. To improve performance, VNE algorithms need to autonomously detect and adapt to complex and time-varying network conditions. They should dynamically provide solutions that are best suited to the current network status, considering factors such as resource availability, network topology, and service requirements.

In this context, Yan et al. [94] address the challenge of automatic virtual network embedding by combining DRL with a neural network structure based on GCN. They propose an algorithm that leverages a parallel RL framework and a multi-objective reward function for training. Through extensive simulations under various scenarios, the algorithm demonstrates strong performance across multiple metrics compared to existing state-of-the-art solutions. Notably, it achieves significant improvements in acceptance ratio and average revenue, with up to 39.6% and 70.6% improvements, respectively. The results also highlight the robustness of the proposed solution in handling different network conditions.

Similarly, Sun et al. [95] introduce *DeepOpt*, a framework where both DRL and GNN are employed to tackle the Virtual network function (VNF) placement challenge. They

formulate the task as a sequential decision-making problem with the aim of minimizing the cost of deployment and enhancing the Quality of service (QoS) of network flows, particularly the E2E delay. They also argue that GNNs contribute to improved generalization across diverse network topologies. The simulation results indeed show that the model outperforms existing algorithms without GNN in the VNF placement task.

To ensure QoS while minimizing processing resources, network operators face the challenge of efficiently migrating traffic flows among network function instances in response to unpredictable network traffic. Sun et al. [96] propose *DeepMigration*, a solution that combines DRL with GNN to dynamically and accurately migrate traffic flows among network function instances (Figure 6). *DeepMigration* leverages the graph-based relationship deduction capability of their customized GNN and the self-evolution ability of DRL to model the cost and benefit of flow migration such as migration latency and reduction in the number of NF instances. This enables *DeepMigration* to generate effective and dynamic flow migration policies.

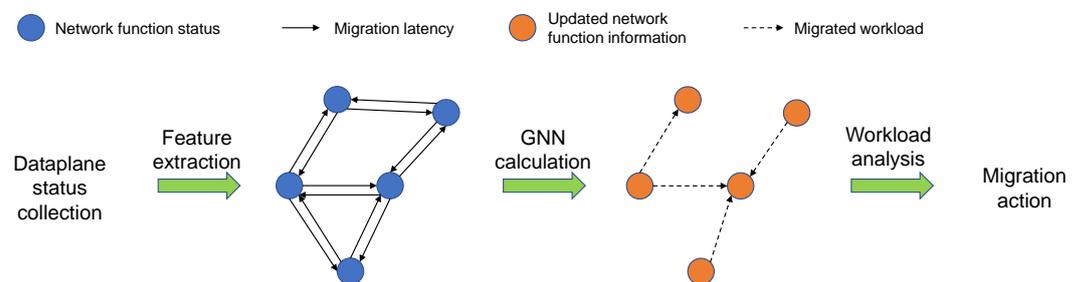


Figure 6. A calculation example of GNN [96].

Habibi et al. [97] address the limitations of previous VNE approaches by introducing GraphViNE, a parallelizable VNE solution based on spatial GNNs. By incorporating server clustering, GraphViNE improves runtime and performance. Simulation experiments demonstrate that GraphViNE reduces runtime by a factor of eight and achieves an approximately 18% improvement in the revenue-to-cost ratio compared to other algorithms.

In a similar direction, Zhang et al. [98] propose a novel VNE algorithm that integrates RL and GNNs. This algorithm leverages a self-defined fitness matrix and fitness value to establish an objective function, enabling the efficient implementation of a dynamic VNE algorithm and effectively reducing resource fragmentation. The proposed method is evaluated using comparison algorithms, and simulation experiments validate its effectiveness. The results demonstrate that the dynamic VNE algorithm, which combines RL and GNNs, exhibits favorable VNE characteristics. Additionally, by modifying the resource attributes of both the physical and virtual networks, the algorithm demonstrates its flexibility in adapting to different network configurations.

Software-defined network (SDN) and Network function virtualization (NFV) have significantly advanced software-based control technology and reduced costs. SFC plays a crucial role in efficiently routing traffic through network servers to process the requested NFV while maintaining Service-Level Agreements (SLAs). However, SFC faces challenges in maintaining high QoS in complex scenarios. Existing approaches, such as deep neural networks (DNNs) [101], lack efficiency in utilizing network topology information and cannot handle dynamically changing topologies. To address these limitations, Heo et al. [99] propose a novel SFC model based on GNNs that leverages the graph-structured characteristics of network topology. Comprising an encoder and a decoder, the model functions by having the encoder capture the representation of the network topology, while the decoder estimates probabilities for neighboring nodes to handle a VNF. Experimental results demonstrate that the proposed GNN-based architecture outperforms previous DNN-based baseline models. Furthermore, the GNN-based model offers flexibility in adapting to new network topologies without requiring redesigning and retraining.

With the increasing demand for dynamic SFCs and the growing sensitivity of service providers towards energy consumption in NFV, there is a need to address the Energy-efficient Graph-structured SFC problem (EG-SFC). Qi et al. [100] recognize this problem and formulate it as a combinatorial optimization problem. To tackle EG-SFC, the authors propose an E2E GNN based on the constrained DRL approach. Their method utilizes GCNs to represent the Q-network of the Double Deep Q-Network (DDQN) [102] within the DRL framework. Additionally, the authors introduce a masking mechanism to mask the servers with insufficient remaining resources by assigning them with large negative values so that they are completely ignored from the deployment of new VNFs. This is expected to accelerate the training process. Experimental results demonstrate the effectiveness of the proposed method in handling unseen SFC graphs. It outperforms the least delay greedy (LDG) algorithms and traditional DDQN approaches, showcasing its ability to achieve better performance in terms of energy efficiency.

3.3.2. Routing

Routing optimization is a critical challenge in computer networks, significantly impacting network performance. The traditional routing protocols such as the Open Shortest Path First (OSPF) [103] and the Broader Gateway Protocol (BGP) [104] are performed on individual network devices based on only the local information. In contrast, SDN with its centralized controller offers a holistic network view, revolutionizing routing optimization by considering entire network configurations and topology.

However, the routing optimization in SDN remains a challenging problem cause of the intractability of estimating the per-path E2E metrics (i.e., delay, jitters, and so on). Analytical models and packet-level simulators are classical solutions to this problem. While the analytical network models such as queueing theory and fluid models are constrained by unrealistic hypotheses, the packet-level simulators can accurately estimate those network metrics but at expensive computing costs. Data-driven methods, and especially those with GNN as the core module (Table 6), have emerged as an alternative due to their scalability and their capability of accurate modeling. In this section, we review two classes of GNN-based solutions: one focuses only on the estimation of the E2E metrics, and the other concerns the E2E DL architectures involved in the decision-making in plus.

Table 6. Summary of application of GNN-based models for routing in core networks.

Ref.	Model	Baseline	Simulator	Metric
[105–110]	MPNN	Queueing theory, Fluid model, RouteNet	OMNeT++	Delays in queuing networks
[111]	MPNN	Genetic Algorithm	-	Edge utilization
[112]	MPNN, DRL	Equal-cost multi-path, DRL [113,114]	OMNeT++	Average E2E delay
[115]	MPNN, DRL	Shortest path, Q-Routing [116]	Mininet, Ryu	Packet delivery ratio and transmission delay
[117]	GCN, DRL	OSPF, DRL [118]	OMNeT++	Packet loss rate, average delay, total number of packets forwarded

On the one hand, Rusek and Cholda [105] first propose the adoption of the MPNN for the estimation of delays in queuing networks. The framework is further improved with a bidirectional message passing between links–paths, so-called RouteNet [106]. The experiments are also carried out with the data generated by the packet-level simulator OMNeT++ [119] providing more realistic network traffic. Since then, many RouteNet variants have been proposed to improve the model and to adapt to the scenarios of more sophisticated network configurations. Badia-Sampera et al. [107] focus on the capability of modeling the network where forward devices would have different queue sizes. Suárez-Varela et al. [108] evaluate the generalization capabilities of the RouteNet in the larger networks of variable size (up to 50 nodes). TwinNet [109] introduces the notion of a queue state which led to a more complex message-passing scheme of three stages: (i) passing messages from queues, links to the associated paths, (ii) passing messages from

paths to queues and (iii) passing messages from paths to links. This enables the model to capture different queue scheduling policies in a real-world network scenario. The most recent variant, RouteNet-F [110], attempts to fill the gap between the scale of the network testbeds or simulations, which are used for training, and the scale of the real network in the deployment. Indeed, this model aims to tackle the two challenges of RouteNet: (i) scaling to larger link capacities and (ii) different output ranges. The former challenges the neural networks to encode out-of-distribution numerical values of link capacities, which can reach infinity. The latter occurs when deploying the trained model into the real larger networks where the delay/jitter values can be very different due to higher link capacities and longer routing paths. Accordingly, the paper proposes to encode the link load in place of link capacity and to aggregate the link-wise effective queue occupancy along a path to infer the delay/jitter.

On the other hand, Sawada et al. [111] aimed to perform routing optimization for a network whose traffic could change every second. The authors propose a GNN-based framework that takes as input the network states (e.g., link bandwidths and traffic demands) and the network topology to generate the routing table. The generated routing tables are then post-processed to guarantee the reachability of the routing path. The model is optimized using the supervised training scheme. The authors apply the Genetic Algorithm onto the given networks to produce the sub-optimal routing tables, which are later used as the ground-truth labels for the training phase. Chen et al. [112] propose a MPNN-based framework for optimizing routing, a so-called AutoGNN. Given the network information as the input, this model combines the graph structural information with the local link information to generate a weight for each link as the output. A weighted shortest path algorithm is then performed on the weighted graph to calculate the shortest path, thus providing automatic routing optimization. The AutoGNN is trained to reduce the overall transmission delay using a Reinforcement learning (RL) mechanism, so-called REINFORCE [120], coupled with an OMNet++ simulator for providing network traffic data. The above methods are centralized in nature, aiming to optimize the routing decisions from a global perspective.

In contrast, Swaminathan et al. [115] propose an MPNN-based model to perform the routing optimization locally. The model, as the core module of a controller, provides the next node for each routing request received on individual nodes. The authors propose the generation of a network state matrix, which is the input for the MPNN-based model. The entire model is trained using the DRL framework which was aimed to maximize the rewards composed of (i) Packet Delivery reward, ensuring that the routing algorithm selects nodes in the neighbor of the current node, and (ii) Delay reward again composing of Queueing delay and Link delay. The training environment is set up using a Mininet Simulator to simulate a SDN and Ryu Software to simulate controller behavior. Similarly, Huang et al. [117] propose a distributed traffic routing control algorithm based on a deep graph reinforcement learning framework that combines the GCN model with a DRL training scheme called Actor-Critic [121]. This framework leverages a GCN model to extract the structural information of the network topology and then generates the next hop policy for each routing request received on individual nodes. The experiments are conducted within an environment interacting with an OMNet++ simulator. It is shown that the framework is capable of reducing packet transmission delay, increasing packet delivery ratio, and reducing the probability of network congestion.

Lesson Learned

GNNs have emerged as a powerful data-driven methodology for network optimization, traffic analysis, and security enhancement. Their applications are diverse, spanning multiple network domains and addressing various fundamental network challenges. Enabled by the holistic view of network topology provided by SDN, GNNs not only optimize networks, but also empower SDN and novel concepts like NFV and network slicing in Next-generation networks. However, the lack of real datasets poses a challenge for bench-

marking and comparing different solutions. Nonetheless, their potential for transforming future networks is undeniable.

4. GNN-Based Digital Twins for Next-Generation Networks

As could be observed previously, Network digital twins (NDTs) offer significant advantages for diverse network applications over multiple network domains (Section 2). Additionally, Graph neural networks (GNNs) are emerging as a promising technique that can be leveraged to solve a wide range of network problems (Section 3). To develop the next-generation communication network with unprecedented capabilities, we identified the Graph neural network-based network digital twin (GraphNDT) as an essential research topic where several problems remain open, and a variety of advantages are to be explored. In addition, we reason on various potential usages of a GraphNDT. In the end, we examine two particular cases in existing works where GraphNDTs are leveraged for enhancing routing and network slicing.

4.1. Major Benefits of Using GNN-Based Digital Twins in Next-Generation Networks

We present below three key advantages of integrating the GNN-based Digital twin (DT) into Next-generation networks (Figure 7). These advantages, although distinct, are not mutually exclusive but rather complement and reinforce each other.

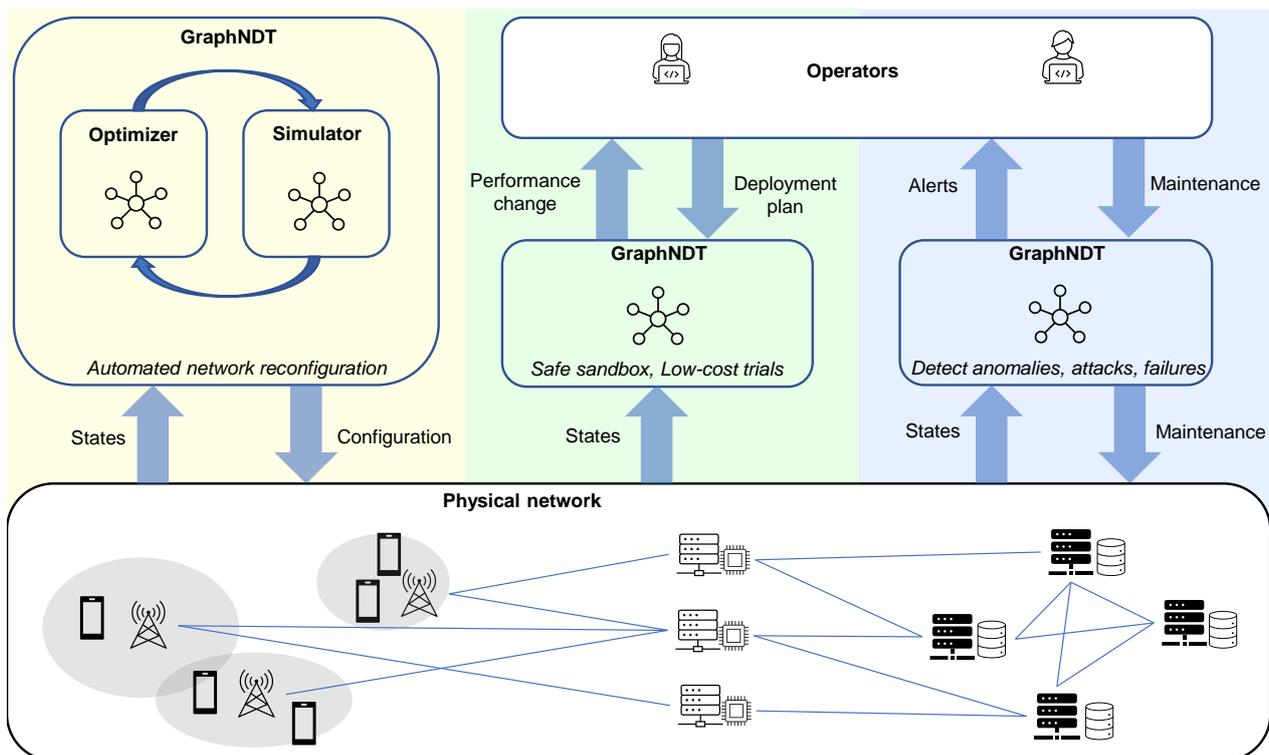


Figure 7. Illustrations of three major use cases of a Graph neural network-based network digital twin (GraphNDT) including (from left to right): network optimization (yellow), low-cost trials (green), and predictive maintenance (blue).

4.1.1. Network Optimization

In the context of a Software-defined network (SDN), the integration of an NDT within the control plane enables dynamic optimization of network operations based on insights gained from the digital twin [34]. The synchronization between network systems and their digital replicas allows for real-time updates and accurate representation of the network state. Moreover, the centralized control plane enables the NDTs to capture the entire network topology, providing the collected data with the semantic relationship between network elements, and allowing the control plane the making of optimal decisions based on a holistic

view of the network. To effectively leverage the graph structure of the network, NDTs can leverage GNNs. GNNs are well-suited for capturing the dependencies and relationships between network elements, as they can learn from the graph structure and features associated with each element. On the one hand, GraphNDTs equipped with optimization algorithms can function as orchestration components within the control plane [34,122]. They utilize network states collected from the synchronization link and leverage GNN models to propose optimal actions based on the gathered information. By analyzing the real-time data from the network using GNN techniques, GraphNDTs can optimize various aspects such as resource allocation, routing decisions, or traffic management to improve network performance (see Section 3). On the other hand, GraphNDTs can also serve as components for validating optimization policies generated by other algorithms [1,109]. Once an optimization policy is generated by an external algorithm, the NDT can simulate its implementation and evaluate its effectiveness within the digital twin environment. By comparing the outcomes of the generated policy with the expected results, the NDT can provide feedback on the efficacy of the optimization policy. This dual functionality of GraphNDTs—acting as optimizers and validators—makes them versatile tools for network optimization.

4.1.2. Low-Cost Trials

With the capability to accurately model a dynamic complex system, DT equipped with data-driven methodologies is a key technology for the exact simulation and cost-effective trials on network systems [1,27,109]. Moreover, since the network is naturally represented as graphs, it is efficient to leverage GNNs for better analyzing the network systems. Provided with the data collected in real time from the physical system, an NDT empowered by GNN-based models can be trained to capture the behavior patterns of the network in diverse situations. This enables the NDT to play a critical role in low-cost trials where operators can interact with the network, try out several combinations of configurations or various scenarios; the NDT can still provide the predicted behaviors of the network with high precision. With generalized GNN models, users can test diverse sets of inputs, including those that could potentially cause service disruptions, which may be impossible to test in a network with zero fault tolerance [109]. These scenarios could involve increasing traffic loads, simulated link failures, deployment of new Base stations (BSs), or network upgrades. In summary, low-cost trials and simulations benefit from accurate modeling provided by digital twins equipped with data-driven methodologies. The integration of GNNs in NDTs enables operators to experiment with different network configurations and scenarios, predicting network behaviors and assessing the impact of various inputs. This capability allows for cost-effective testing of scenarios that may involve service disruptions, helping operators make informed decisions for network optimization and improvement.

4.1.3. Predictive Maintenance

The capabilities of a GraphNDT to analyze system states over time can be leveraged for predictive maintenance, including predicting failure points, tracing back to root causes, and making maintenance decisions [28,31,43,44]. By providing two-way communication, an NDT not only allows synchronizing the digital twin with the actual network infrastructure, but also provides a means to apply remote reconfiguration, simulate potential changes or maintenance activities, and validate their impact on network performance. Additionally, GraphNDT can facilitate the detection of anomalies or attacks by applying data-driven methods to time series data combined with network topology (see Section 3.2.3). After that, the graph structure, representing the communication between network elements, is crucial for performing root cause analysis. GNNs are well-suited for exploiting this structural information and tracing back to the root cause of anomalies detected within the network. Therefore, the combination of an NDT and GNNs empowers predictive maintenance, enabling a comprehensive understanding of network behavior and proactive maintenance decision making.

4.2. Existing Studies

In the following subsections, we showcase two specific scenarios where GraphNDTs help improve network performance. These scenarios exemplify the practical applications of GraphNDTs in addressing routing optimization challenges and enabling effective network slicing management.

4.2.1. Routing Optimization

We present the use cases of TwinNet [109], which is a GraphNDT that introduces a digital twin empowered by a GNN model to accurately simulate networks with various queuing configurations, routing, topologies, and traffic models. TwinNet is trained using data generated from a packet-accurate network simulator aiming at producing accurate estimates of per-flow Quality of service (QoS) metrics (e.g., delay, jitter, loss). TwinNet could be coupled with an optimizer to find the routing policies that satisfy complex Service level agreements (SLAs) with increasing traffic intensity. Moreover, the predictability of TwinNet is demonstrated through a what-if scenario involving a budget-constrained network upgrade, showcasing its ability to assess the impact of network upgrades, enabling low-cost trials.

Accurate modeling of networks is vital to network optimization. Although there exist traditional network models such as queuing theory and fluid model which are extensively used, they often rely on simplifying assumptions and may have limitations in capturing the complexity and dynamics of real-world network systems. Alternatively, packet-level simulators can accurately model networks but come at a high computational cost. In contrast, Deep learning (DL) techniques, particularly GNNs, offer a promising approach that strikes a balance between accuracy and computational cost. GNNs can accurately model network behaviors by being trained on real data without relying on excessive assumptions. Compared to packet-level simulators, GNNs offer a more computationally efficient solution for network modeling. Once trained, GNN-based models can make predictions on network performance in a timely manner.

TwinNet is trained on the dataset generated from packet-level simulator OMNeT++. Each data sample consists of a simulated network scenario defined by a topology, a source-destination traffic matrix, and a routing and queuing policy. The communication is simulated and the metrics of network performance are obtained to label the corresponding data sample. The traffic matrices are generated to simulate a wide range of network states from low to high traffic loads, including those that may cause congestion and packet loss. Queuing configurations are randomly assigned to each node. The training and evaluation of TwinNet utilize three real-world network topologies. The quantitative results demonstrate the effectiveness of TwinNet, achieving a Mean average percentage error (MAPE) of 3.8%, outperforming baselines that include a multi-layered neural network and a RouteNet [106]. Additionally, TwinNet can be combined with an optimization algorithm for routing optimization. In the paper, experimental results from coupling TwinNet with Direct Search showcase superior performance compared to baselines, which involve the combination of fluid models and the shortest path search. Furthermore, TwinNet enables low-cost trials without impacting the real network. Using TwinNet, the authors can assess the network's capacity enhancement by selecting an additional link to be added. The results demonstrate that the additional link suggested by TwinNet effectively reduces the delay by 40.1% to 54.3%.

4.2.2. Network Slicing Management

We examine the application of a GNN-based digital twin for network slicing as proposed by Wang et al. [122]. The developed digital twin focuses on network slicing management, utilizing GNNs operating on a multilayered network constructed from slicing configurations. It is capable of performing the prediction of per-slice End-to-end (E2E) metrics, in which the latency was considered as an example of the target metrics. Moreover, three potential what-if scenarios were considered as the showcases for the leverage of the digital twin.

In 5G networks, network slicing can leverage the virtualized and software-defined nature of the network infrastructure to propose a customized slicing configuration that aligns with the specific usage requirements. This approach brings significant enhancements in terms of QoS. However, achieving this goal is a challenging task primarily due to the difficulty of measuring E2E metrics across multiple network slices spanning multiple network domains and involving different network nodes. Moreover, a digital twin targeting the capability of performing what-if scenarios may require the accurate modeling of the network behaviors given different configurations. To fulfill these requirements, the authors made use of GraphSAGE [85], a GNN to build a data-driven network modeling capable of predicting with precision the E2E latency of multiple network slices given the slicing configurations and substrate topology.

To train the GraphNDT, the authors reused the data from [106]. The dataset was initially generated from OMNeT++ for routing scenarios which consist of three different topologies, routing configurations, traffic matrices, and captured network metrics such as delay, latency, and jitter. However, by converting the pair-wise source destination traffic matrices into network slices, that dataset was used to stimulate the network slicing management scenario, and the ground-truth network metrics represent the E2E network slicing metrics. The experimental results demonstrate the accuracy of the proposed framework where the predicted latency has less than a 5% error rate under all three network topologies. Furthermore, three what-if scenarios are introduced to showcase the advantages of an NDT. Firstly, it can simulate an increase in traffic intensity to observe potential SLA violations, thus providing network operators with insights into the need for network slice reconfiguration. Secondly, it can simulate link failures and evaluate the monitoring system's adaptive capabilities by observing changes in network metrics. Lastly, it can assess the performance of different optimization algorithms by comparing network metrics with the required SLA thresholds. Overall, the combination of the repurposed dataset and the GraphNDT enables accurate simulation and evaluation of network slicing scenarios, allowing for effective network optimization and management.

5. Challenges and Future Directions

We showed that GraphNDT can be a key enabler for innovating communication networks in many aspects including optimization, management, and security. However, the deployment of GraphNDTs may entail overcoming various obstacles. In this regard, we outline some of these challenges along with potential research directions, as summarized in Table 7.

Table 7. Summary of challenges and future directions for GraphNDTs.

Challenges	Description	Future Directions
Dynamicity	The network topology is dynamic in a way that nodes or edges may appear or disappear, and the input data change over time.	<ol style="list-style-type: none"> 1. Incremental Learning can allow GNN-based models learning from new data without having to retrain from scratch, which could be particularly beneficial in dynamic environments where data continually evolve [123]. 2. FL plays a crucial role within the dynamic GNN model, particularly in handling complex, large scales, and adapting to shifts in the network structure over time [124,125].
Heterogeneity	The different types of nodes and edges have different attributes, which are usually located in different feature spaces.	<ol style="list-style-type: none"> 1. Meta-path Aggregated Graph Neural Network (MAGNN) which can capture both the semantic and structural information in heterogeneous graphs. 2. Contrastive learning-based method, where the model is pre-trained in a self-supervised manner to learn both the semantic and structural properties of heterogeneous graphs.
Robustness	The ability of the model to maintain high performance even when faced with perturbations in the graph structure or feature information.	<ol style="list-style-type: none"> 1. Robust GNN architectures that are inherently more robust against adversarial attacks [126,127]. 2. Adversarial training methods involve integrating adversarial examples into the training process [128].
Generalization	The ability to generalize GNNs so that the GraphNDT can still provide accurate predictions in case of abnormal scenarios or configurations. This is mandatory for low-cost trial use cases.	<ol style="list-style-type: none"> 1. Train the models on additional data generated by simulations and testbeds. Transfer learning can be adopted to finetune the pre-trained models on the current NDTs. 2. FL enables collaborative learning and model aggregation, allowing the GraphNDTs to benefit from a broader understanding of various configurations while preserving the privacy of individual network systems [129–131].

5.1. Dynamicity

In real-world scenarios, network topologies exhibit dynamic characteristics with nodes and edges appearing or disappearing, or the input data change over time. Within an NDT where the two-way communication enables the real-time collection of data, the ability to exploit and understand these dynamic network behaviors becomes crucial. Static graphs are stable so they can be modeled feasibly, while dynamic graphs introduce changing structures. Thus, new GNN models are needed to adapt to the dynamicity of the network topology because re-training the model can be computationally expensive. Finding more efficient ways to update node representations in response to graph changes in network topology is an ongoing research problem. In this context, *GraphSAIL* has been proposed for the first time [123]. It addresses the incremental learning on GNN for recommender systems, which deals with the edges between nodes. Incremental learning [132,133] is a method that allows a model learning from new data without having to retrain from scratch, which could be particularly beneficial in dynamic environments where data continually evolve. Also, incorporating other machine learning techniques can enhance the performance of GNNs with dynamic network topology. Integrating FL [130,131,134–136] with dynamic GNNs can be a promising direction for building adaptable models for Next-generation networks. For example, FL allows dynamic GNNs adaptation to changes in data distribution across devices. In particular, using an asynchronous update allows procession of fast nodes without waiting for slower or disconnected ones [125]. Another strategy is to train GNNs with FL while accounting for clients that may drop out during the process [124]. All these can be beneficial for dynamic GNNs with Next-generation networks, where the relationships between nodes may be changing over time, for straggler nodes, added, disconnected, or updated nodes.

5.2. Heterogeneity

Although NDTs can collect network configuration data in the form of graphs, effectively processing these data for training machine learning models, such as GNNs, necessitates thoughtful consideration and meticulous engineering. One of the crucial challenges in this process is identifying the suitable entities to be represented as nodes and determining the relationships between them as edges. Real-world graphs such as network topologies are often heterogeneous, containing various types of nodes and edges, each with their unique attributes and typically situated in different feature spaces. The evolving nature of these graph structures and information makes the application of GNNs to Next-generation networks a complex task. Consequently, careful consideration is required during the preprocessing step to accommodate graph heterogeneity. Nevertheless, most existing GNN-based models primarily handle homogeneous graphs with a singular node type and edge type.

Recently, innovative solutions have emerged to address these issues; for example, studies such as those proposed in [137,138]. In particular, the authors in [137] proposed a new model named *MAGNN*, which employed three major components, (i) the node content transformation to encapsulate input node attributes, (ii) the intra-meta path aggregation to incorporate intermediate semantic nodes, (iii) and the inter-meta path aggregation to combine messages from multiple meta paths. Experimental results demonstrate that *MAGNN* can significantly outperform state-of-the-art models on various heterogeneous graph mining tasks, including node classification, link prediction, and recommendation. The results indicate that the model successfully captures both the semantic and structural information in heterogeneous graphs, demonstrating the potential for handling complex real-world graph data. Moreover, the work in [138] proposed an innovative Contrastive Pre-Training strategy for GNNs on Heterogeneous Graphs (CPTHG). This strategy captures both the semantic and structural properties of graphs in a self-supervised manner. Contrastive learning is a form of self-supervised learning where the model is trained to identify which samples in the dataset are similar (or contrastive) to each other. The proposed model contains three main components, (i) a semantic-preserving contrast that aims

to maximize the similarity between the node pairs that have the same type or share the same semantic information, (ii) a structure-preserving contrast that helps to maximize the similarity between the node pairs that have similar structural roles in the network, and (iii) negative sampling to reduce the similarity between the nodes that do not share similar structural roles or semantic information. These advancements suggest a promising direction for future work involving these techniques, particularly in the domain of network traffic prediction.

5.3. Robustness

GNN-based solutions need to maintain high performance even when faced with perturbations in the graph structure or feature information. Adversarial attacks pose a particular concern, where small, intentionally harmful changes are made to the graph (such as adding, removing edges, or altering node features), or simply dealing with noisy or incomplete data. Within an NDT system, this attack can be performed by exploiting the vulnerabilities in the communication channels between the network and the digital replica. For instance, if data collection involves transmitting data over a network, attackers could attempt to intercept and modify the data packets, altering network statistics, topology information, or node attributes. Adversarial attacks significantly impact the integrity and reliability of GraphNDT systems and can lead to incorrect predictions or interpretations by GNN models. For example, in a semi-supervised node classification task, adversarial modifications can make a GNN misclassify the label of a node. The authors in [139] highlighted this problem, where they demonstrated that most GNNs are highly vulnerable to adversarial attacks.

In addition to securing the communication channels, enhancing the robustness of GNN models against adversarial attacks is an intrinsic solution. This can be achieved through the development of robust GNN architectures and the application of adversarial training techniques. For robust GNN architectures, some research papers have designed new GNN architectures that are inherently more robust against adversarial attacks. For example, *RGCN (Robust Graph Convolutional Network)* incorporates a smoothing mechanism to reduce the effect of adversarial perturbations [127]. In the same direction, a *GNN-Guard* against adversarial attacks on GNN was proposed in [126]. The main idea is to monitor the information flow in a GNN during message passing and to control the influence of nodes that have irregular behavior. To achieve this, significant modification is involved in the way GNN aggregates and propagates information. This includes introducing guarded aggregation to limit the influence of suspicious nodes and guarded influence computation to attenuate the impact of adversarial nodes. On the other hand, adversarial training methods involve integrating adversarial examples into the training process. By training the model on both original and adversarial data, GNN models learn to improve their performance in the presence of adversarial perturbations [128]. To further mitigate the effects of some types of adversarial attacks, adding regularization terms to the loss function of the model can help improve the model's robustness by discouraging complex models that could overfit the perturbed data.

5.4. Generalization

One of the key challenges in deploying GraphNDTs for cost-effective trials in networks is ensuring their generalizability across various scenarios and configurations, including disruptive events that may lead to service disruptions. In zero-tolerance systems, exploring these disruptive scenarios is impossible. As a data-driven approach, GraphNDTs rely on training data to make accurate predictions, but without access to data from such scenarios, they may struggle to accurately predict and simulate disruptions in networking systems.

To address this challenge, two potential solutions can enhance the generalizability of GraphNDTs. On the one hand, we can train GraphNDTs on additional data generated from simulations or testbeds that simulate a wide range of network scenarios and configurations [109,110,140]. By being exposed to diverse simulated disruptive events,

GraphNDTs can learn to recognize patterns and dynamics associated with service disruptions. Training on additional data helps GraphNDTs capture the complexities of disruptive scenarios and improves their ability to generalize to unseen disruptions in zero-tolerance systems. After that, we can leverage transfer learning techniques to finetune the pre-trained GraphNDTs to the actual network systems being twinned. This fine-tuning process allows GraphNDT ability to adapt its predictions to the specific dynamics and characteristics of the target system, leveraging the prior knowledge acquired during training on the additional data. On the other hand, the aggregation of different GraphNDTs can also contribute to generalization by capturing the knowledge of the same application (e.g., routing, network slicing) across different networking systems. However, personalization becomes a key challenge due to the varying characteristics of different networking systems, which can be addressed by employing FL techniques. FL enables collaborative learning and model aggregation, preserving the privacy of individual network systems [129–131] while allowing GraphNDTs benefit from a broader understanding of various configurations.

6. Conclusions

In this article, we delved into the vast possibilities afforded by GNNs and DT for upcoming network generations. This work was motivated by the lack of a comprehensive survey on the use of GNN and DT within Next-generation networks. To bridge this gap, we first provided an updated survey on the application of DT and GNN for Next-generation networks. Then, we introduced the recent advances and integration of GNN within DT accompanied by highlighting its principal benefits and case studies. Finally, we identified key research challenges and possible directions for further investigation. We believe that this article will stimulate more attention in this emerging area and encourage more research efforts toward the realization of GNN with digital twins.

Author Contributions: Conceptualization, D.-T.N., O.A. and K.P.; Writing—Original Draft, D.-T.N., O.A., K.P., T.H. and P.R.-P.; Writing—Review and Editing, D.-T.N., O.A., K.P., T.H. and P.R.-P.; Visualization, D.-T.N. and O.A.; Supervision, K.P., T.H. and P.R.-P. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: No new data were created or analyzed in this study. Data sharing is not applicable to this article.

Conflicts of Interest: Author Duc-Thinkh Ngo is a employee of company (Orange Innovation); Author Thomas Hassan is a employee of company (Orange Innovation); Author Philippe Raipin-Parvédy is a employee of company (Orange Innovation); Author Ons Aouedi is a employee of company (Kandaraj Piamrat); The authors declare no conflicts of interest.

Abbreviations

The following abbreviations are used in this manuscript:

CSI	Channel state information
LSTM	Long short-term memory
GNN	Graph neural network
GCN	Graph convolutional network
CNN	Convolutional neural network
RNN	Recurrent neural network
GSP	Graph signal processing
GAT	Graph attention network
MAPE	Mean average percentage error
GRU	Gated recurrent unit
FL	Federated learning
URLLC	Ultra-reliable low-latency communication
UE	User equipment
SLA	Service level agreement

E2E	End-to-end
NDT	Network digital twin
DT	Digital twin
DRL	Deep reinforcement learning
RL	Reinforcement learning
MARL	Multi-agent reinforcement learning
BS	Base station
RAN	Radio access network
VNE	Virtual network embedding
SFC	Service function chain
VNF	Virtual network function
NFV	Network function virtualization
DL	Deep learning
SDN	Software-defined network
MPNN	Message-passing neural network
MEC	Multiple-access edge computing
IoT	Internet of things
IIoT	Industrial internet of things
ML	Machine learning
DDoS	Distributed denial of service
YANG	Yet another next generation
QoS	Quality of service
GFT	Graph fourier transformation
GraphNDT	Graph neural network-based network digital twin
AI	Artificial intelligence
NGN	Next-generation network

References

- Almasan, P.; Ferriol-Galmes, M.; Paillisse, J.; Suarez-Varela, J.; Perino, D.; Lopez, D.; Perales, A.A.P.; Harvey, P.; Ciavaglia, L.; Wong, L.; et al. Network Digital Twin: Context, Enabling Technologies, and Opportunities. *IEEE Commun. Mag.* **2022**, *60*, 22–27. [CrossRef]
- Zhou, C.; Yang, H.; Duan, X.; Lopez, D.; Pastor, A.; Wu, Q.; Boucadair, M.; Jacquenet, C. Digital Twin Network: Concepts and Reference Architecture. 2023. Available online: <https://datatracker.ietf.org/doc/draft-irtf-nmrg-network-digital-twin-arch/04/> (accessed on 1 October 2023).
- Kuruvatti, N.P.; Habibi, M.A.; Partani, S.; Han, B.; Fellan, A.; Schotten, H.D. Empowering 6G Communication Systems with Digital Twin Technology: A Comprehensive Survey. *IEEE Access* **2022**, *10*, 112158–112186. [CrossRef]
- Jiang, W. Graph-Based Deep Learning for Communication Networks: A Survey. *Comput. Commun.* **2022**, *185*, 40–54. [CrossRef]
- Vesselinova, N.; Steinert, R.; Perez-Ramirez, D.F.; Boman, M. Learning Combinatorial Optimization on Graphs: A Survey with Applications to Networking. *IEEE Access* **2020**, *8*, 120388–120416. [CrossRef]
- Zhu, Y.; Du, Y.; Wang, Y.; Xu, Y.; Zhang, J.; Liu, Q.; Wu, S. A survey on deep graph generation: Methods and applications. *arXiv* **2022**, arXiv:2203.06714.
- He, S.; Xiong, S.; Ou, Y.; Zhang, J.; Wang, J.; Huang, Y.; Zhang, Y. An overview on the application of graph neural networks in wireless networks. *IEEE Open J. Commun. Soc.* **2021**, *2*, 2547–2565. [CrossRef]
- Suárez-Varela, J.; Almasan, P.; Ferriol-Galmés, M.; Rusek, K.; Geyer, F.; Cheng, X.; Shi, X.; Xiao, S.; Scarselli, F.; Cabellos-Aparicio, A.; et al. Graph Neural Networks for Communication Networks: Context, Use Cases and Opportunities. *arXiv* **2022**, arXiv:2112.14792.
- Yuan, H.; Yu, H.; Gui, S.; Ji, S. Explainability in graph neural networks: A taxonomic survey. *IEEE Trans. Pattern Anal. Mach. Intell.* **2022**, *45*, 5782–5799. [CrossRef]
- Skarding, J.; Gabrys, B.; Musial, K. Foundations and modeling of dynamic networks using dynamic graph neural networks: A survey. *IEEE Access* **2021**, *9*, 79143–79168. [CrossRef]
- Abadal, S.; Jain, A.; Guirado, R.; López-Alonso, J.; Alarcón, E. Computing graph neural networks: A survey from algorithms to accelerators. *ACM Comput. Surv. (CSUR)* **2021**, *54*, 1–38. [CrossRef]
- Dong, G.; Tang, M.; Wang, Z.; Gao, J.; Guo, S.; Cai, L.; Gutierrez, R.; Campbel, B.; Barnes, L.E.; Boukhechba, M. Graph neural networks in IoT: A survey. *ACM Trans. Sens. Netw.* **2023**, *19*, 1–50. [CrossRef]
- Jiang, W.; Luo, J. Graph neural network for traffic forecasting: A survey. *Expert Syst. Appl.* **2022**, *207*, 117921. [CrossRef]
- Nguyen, H.X.; Trestian, R.; To, D.; Tatipamula, M. Digital Twin for 5G and Beyond. *IEEE Commun. Mag.* **2021**, *59*, 10–15. [CrossRef]

15. Wu, Y.; Zhang, K.; Zhang, Y. Digital twin networks: A survey. *IEEE Internet Things J.* **2021**, *8*, 13789–13804. [[CrossRef](#)]
16. Khan, L.U.; Han, Z.; Saad, W.; Hossain, E.; Guizani, M.; Hong, C.S. Digital Twin of Wireless Systems: Overview, Taxonomy, Challenges, and Opportunities. *arXiv* **2022**, arXiv:2202.02559.
17. Suhail, S.; Hussain, R.; Jurdak, R.; Oracevic, A.; Salah, K.; Hong, C.S.; Matulevičius, R. Blockchain-based digital twins: Research trends, issues, and future challenges. *ACM Comput. Surv. (CSUR)* **2022**, *54*, 1–34. [[CrossRef](#)]
18. Tao, F.; Cheng, J.; Qi, Q.; Zhang, M.; Zhang, H.; Sui, F. Digital twin-driven product design, manufacturing and service with big data. *Int. J. Adv. Manuf. Technol.* **2018**, *94*, 3563–3576. [[CrossRef](#)]
19. Coupaye, T.; Bolle, S.; Derrien, S.; Folz, P.; Meye, P.; Privat, G.; Raipin-Parvedy, P. *A Graph-Based Cross-Vertical Digital Twin Platform for Complex Cyber-Physical Systems*; Springer International Publishing: Cham, Switzerland, 2023; pp. 337–363. [[CrossRef](#)]
20. Chen, X.; Kang, E.; Shiraishi, S.; Preciado, V.M.; Jiang, Z. Digital Behavioral Twins for Safe Connected Cars. In Proceedings of the 21th ACM/IEEE International Conference on Model Driven Engineering Languages and Systems, MODELS'18, New York, NY, USA, 14–19 October 2018; pp. 144–153. [[CrossRef](#)]
21. Kumar, S.A.; Madhumathi, R.; Chelliah, P.R.; Tao, L.; Wang, S. A novel digital twin-centric approach for driver intention prediction and traffic congestion avoidance. *J. Reliab. Intell. Environ.* **2018**, *4*, 199–209. [[CrossRef](#)]
22. Tripathy, A.K.; Tripathy, P.K.; Mohapatra, A.G.; Ray, N.K.; Mohanty, S.P. WeDoShare: A Ridesharing Framework in Transportation Cyber-Physical System for Sustainable Mobility in Smart Cities. *IEEE Consum. Electron. Mag.* **2020**, *9*, 41–48. [[CrossRef](#)]
23. Conejos Fuertes, P.; Martínez Alzamora, F.; Carot, M.H.; Campos, J.C.A. Building and exploiting a Digital Twin for the management of drinking water distribution networks. *Urban Water J.* **2020**, *17*, 704–713. [[CrossRef](#)]
24. Mortlock, T.; Muthirayan, D.; Yu, S.Y.; Khargonekar, P.P.; Abdullah Al Faruque, M. Graph Learning for Cognitive Digital Twins in Manufacturing Systems. *IEEE Trans. Emerg. Top. Comput.* **2022**, *10*, 34–45. [[CrossRef](#)]
25. *Digital Twin Network—Requirements and Architecture*; Recommendation ITU-T Y.3090; International Telecommunication Union: Geneva, Switzerland, 2022.
26. Pengnoo, M.; Barros, M.T.; Wuttisittikulij, L.; Butler, B.; Davy, A.; Balasubramaniam, S. Digital Twin for Metasurface Reflector Management in 6G Terahertz Communications. *IEEE Access* **2020**, *8*, 114580–114596. [[CrossRef](#)]
27. Jiang, S.; Alkhateeb, A. Digital Twin Based Beam Prediction: Can We Train in the Digital World and Deploy in Reality? *arXiv* **2023**, arXiv:2301.07682.
28. Akbarian, F.; Fitzgerald, E.; Kihl, M. Intrusion Detection in Digital Twins for Industrial Control Systems. In Proceedings of the 2020 International Conference on Software, Telecommunications and Computer Networks (SoftCOM), Split, Croatia, 17–19 September 2020; pp. 1–6. [[CrossRef](#)]
29. Eckhart, M.; Ekelhart, A. Towards Security-Aware Virtual Environments for Digital Twins. In Proceedings of the 4th ACM Workshop on Cyber-Physical System Security, Incheon, Republic of Korea, 4 June 2018.
30. Eckhart, M.; Ekelhart, A. A Specification-Based State Replication Approach for Digital Twins. In Proceedings of the 2018 Workshop on Cyber-Physical Systems Security and Privacy, CPS-SPC'18, New York, NY, USA, 15–19 October 2018; pp. 36–47. [[CrossRef](#)]
31. Benedictis, A.D.; Flammini, F.; Mazzocca, N.; Somma, A.; Vitale, F. Digital Twins for Anomaly Detection in the Industrial Internet of Things: Conceptual Architecture and Proof-of-Concept. *IEEE Trans. Ind. Inform.* **2023**, *19*, 11553–11563. [[CrossRef](#)]
32. Kephart, J.; Chess, D. The vision of autonomic computing. *Computer* **2003**, *36*, 41–50. [[CrossRef](#)]
33. Hui, Y.; Zhao, G.; Li, C.; Cheng, N.; Yin, Z.; Luan, T.H.; Xiao, X. Digital Twins Enabled On-Demand Matching for Multi-Task Federated Learning in HetVNs. *IEEE Trans. Veh. Technol.* **2023**, *72*, 2352–2364. [[CrossRef](#)]
34. Zhao, L.; Han, G.; Li, Z.; Shu, L. Intelligent digital twin-based software-defined vehicular networks. *IEEE Netw.* **2020**, *34*, 178–184. [[CrossRef](#)]
35. Lu, Y.; Maharjan, S.; Zhang, Y. Adaptive Edge Association for Wireless Digital Twin Networks in 6G. *IEEE Internet Things J.* **2021**, *8*, 16219–16230. [[CrossRef](#)]
36. Van Huynh, D.; Nguyen, V.D.; Sharma, V.; Dobre, O.A.; Duong, T.Q. Digital Twin Empowered Ultra-Reliable and Low-Latency Communications-based Edge Networks in Industrial IoT Environment. In Proceedings of the ICC 2022—IEEE International Conference on Communications, Seoul, Republic of Korea, 16–20 May 2022; pp. 5651–5656. [[CrossRef](#)]
37. Do-Duy, T.; Van Huynh, D.; Dobre, O.A.; Canberk, B.; Duong, T.Q. Digital Twin-Aided Intelligent Offloading with Edge Selection in Mobile Edge Computing. *IEEE Wirel. Commun. Lett.* **2022**, *11*, 806–810. [[CrossRef](#)]
38. Duong, T.Q.; Van Huynh, D.; Li, Y.; Garcia-Palacios, E.; Sun, K. Digital Twin-enabled 6G Aerial Edge Computing with Ultra-Reliable and Low-Latency Communications: (Invited Paper). In Proceedings of the 2022 1st International Conference on 6G Networking (6GNet), Paris, France, 6–8 July 2022; pp. 1–5. [[CrossRef](#)]
39. Wang, D.; Zhang, Z.; Zhang, M.; Fu, M.; Li, J.; Cai, S.; Zhang, C.; Chen, X. The Role of Digital Twin in Optical Communication: Fault Management, Hardware Configuration, and Transmission Simulation. *IEEE Commun. Mag.* **2021**, *59*, 133–139. [[CrossRef](#)]
40. Cho, K.; van Merriënboer, B.; Gülçehre, Ç.; Bahdanau, D.; Bougares, F.; Schwenk, H.; Bengio, Y. Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation. In Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, Doha, Qatar, 25–29 October 2014, a Meeting of SIGDAT, a Special Interest Group of the ACL; Moschitti, A., Pang, B., Daelemans, W., Eds.; Association for Computational Linguistics: Kerrville, TX, USA, 2014; pp. 1724–1734. [[CrossRef](#)]

41. Chen, T.; Guestrin, C. XGBoost: A Scalable Tree Boosting System. In Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'16), San Francisco, CA, USA, 13–17 August 2016; pp. 785–794. [[CrossRef](#)]
42. Hochreiter, S.; Schmidhuber, J. Long Short-Term Memory. *Neural Comput.* **1997**, *9*, 1735–1780. [[CrossRef](#)] [[PubMed](#)]
43. Seilov, S.Z.; Kuzbayev, A.T.; Seilov, A.A.; Shyngisov, D.S.; Goikhman, V.Y.; Levakov, A.K.; Sokolov, N.A.; Zhursinbek, Y.S. The Concept of Building a Network of Digital Twins to Increase the Efficiency of Complex Telecommunication Systems. *Complexity* **2021**, *2021*, 9480235. [[CrossRef](#)]
44. Yigit, Y.; Bal, B.; Karameseoglu, A.; Duong, T.Q.; Canberk, B. Digital Twin-Enabled Intelligent DDoS Detection Mechanism for Autonomous Core Networks. *IEEE Commun. Stand. Mag.* **2022**, *6*, 38–44. [[CrossRef](#)]
45. Ortega, A.; Frossard, P.; Kovačević, J.; Moura, J.M.F.; Vandergheynst, P. Graph Signal Processing: Overview, Challenges, and Applications. *Proc. IEEE* **2018**, *106*, 808–828. [[CrossRef](#)]
46. Defferrard, M.; Bresson, X.; Vandergheynst, P. Convolutional Neural Networks on Graphs with Fast Localized Spectral Filtering. *Proc. Adv. Neural Inf. Process. Syst.* **2016**, *29*, 3844–3852.
47. Li, Y.; Yu, R.; Shahabi, C.; Liu, Y. Diffusion Convolutional Recurrent Neural Network: Data-Driven Traffic Forecasting. In Proceedings of the International Conference on Learning Representations (ICLR'18), Vancouver, BC, Canada, 30 April–3 May 2018.
48. Kipf, T.N.; Welling, M. Semi-Supervised Classification with Graph Convolutional Networks. In Proceedings of the International Conference on Learning Representations (ICLR), Toulon, France, 24–26 April 2017.
49. Gilmer, J.; Schoenholz, S.S.; Riley, P.F.; Vinyals, O.; Dahl, G.E. Neural Message Passing for Quantum Chemistry. In Proceedings of the 34th International Conference on Machine Learning—ICML'17, Sydney, Australia, 6–11 August 2017; Volume 70, pp. 1263–1272.
50. Veličković, P.; Cucurull, G.; Casanova, A.; Romero, A.; Liò, P.; Bengio, Y. Graph Attention Networks. In Proceedings of the International Conference on Learning Representations, Vancouver, BC, Canada, 30 April–3 May 2018.
51. Maas, A.L.; Hannun, A.Y.; Ng, A.Y. Rectifier Nonlinearities Improve Neural Network Acoustic Models. In Proceedings of the 30th International Conference on Machine Learning, Atlanta, GA, USA, 16–21 June 2013; Volume 28.
52. Vaswani, A.; Shazeer, N.M.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, L.; Polosukhin, I. Attention is All you Need. *Adv. Neural Inf. Process. Syst.* **2016**, *30*, 5998–6008.
53. Zhou, J.; Cui, G.; Hu, S.; Zhang, Z.; Yang, C.; Liu, Z.; Wang, L.; Li, C.; Sun, M. Graph neural networks: A review of methods and applications. *AI Open* **2020**, *1*, 57–81. [[CrossRef](#)]
54. Xu, K.; Hu, W.; Leskovec, J.; Jegelka, S. How Powerful are Graph Neural Networks? In Proceedings of the International Conference on Learning Representations, New Orleans, LA, USA, 6–9 May 2019.
55. Alon, U.; Yahav, E. On the Bottleneck of Graph Neural Networks and its Practical Implications. In Proceedings of the International Conference on Learning Representations, Virtual, 3–7 May 2021.
56. Rampášek, L.; Galkin, M.; Dwivedi, V.P.; Luu, A.T.; Wolf, G.; Beaini, D. Recipe for a General, Powerful, Scalable Graph Transformer. *Adv. Neural Inf. Process. Syst.* **2022**, *35*, 14501–14515.
57. Orhan, O.; Swamy, V.N.; Tetzlaff, T.; Nassar, M.; Nikopour, H.; Talwar, S. Connection Management xAPP for O-RAN RIC: A Graph Neural Network and Reinforcement Learning Approach. In Proceedings of the 2021 20th IEEE International Conference on Machine Learning and Applications (ICMLA), Pasadena, CA, USA, 13–16 December 2021; pp. 936–941. [[CrossRef](#)]
58. Zhao, D.; Qin, H.; Song, B.; Han, B.; Du, X.; Guizani, M. A Graph Convolutional Network-Based Deep Reinforcement Learning Approach for Resource Allocation in a Cognitive Radio Network. *Sensors* **2020**, *20*, 5216. [[CrossRef](#)]
59. Shao, Y.; Li, R.; Hu, B.; Wu, Y.; Zhao, Z.; Zhang, H. Graph Attention Network-Based Multi-Agent Reinforcement Learning for Slicing Resource Management in Dense Cellular Network. *IEEE Trans. Veh. Technol.* **2021**, *70*, 10792–10803. [[CrossRef](#)]
60. Hou, K.; Xu, Q.; Zhang, X.; Huang, Y.; Yang, L. User Association and Power Allocation Based on Unsupervised Graph Model in Ultra-Dense Network. In Proceedings of the 2021 IEEE Wireless Communications and Networking Conference (WCNC), Nanjing, China, 29 March–1 April 2021; pp. 1–6. [[CrossRef](#)]
61. Lee, M.; Yu, G.; Li, G.Y. Wireless Link Scheduling for D2D Communications with Graph Embedding Technique. In Proceedings of the ICC 2020—2020 IEEE International Conference on Communications (ICC), Dublin, Ireland, 7–11 June 2020; pp. 1–6. [[CrossRef](#)]
62. Zhao, Z.; Verma, G.; Swami, A.; Segarra, S. Delay-Oriented Distributed Scheduling Using Graph Neural Networks. In Proceedings of the ICASSP 2022—2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Singapore, 23–27 May 2022; pp. 8902–8906. [[CrossRef](#)]
63. Zhang, K.; Yang, Z.; Başar, T. Multi-Agent Reinforcement Learning: A Selective Overview of Theories and Algorithms. In *Handbook of Reinforcement Learning and Control*; Vamvoudakis, K.G., Wan, Y., Lewis, F.L., Cansever, D., Eds.; Springer International Publishing: Cham, Switzerland, 2021; pp. 321–384. [[CrossRef](#)]
64. Mnih, V.; Kavukcuoglu, K.; Silver, D.; Rusu, A.A.; Veness, J.; Bellemare, M.G.; Graves, A.; Riedmiller, M.A.; Fidjeland, A.K.; Ostrovski, G.; et al. Human-level control through deep reinforcement learning. *Nature* **2015**, *518*, 529–533. [[CrossRef](#)]
65. Sutton, R.S.; McAllester, D.; Singh, S.; Mansour, Y. Policy Gradient Methods for Reinforcement Learning with Function Approximation. In *Proceedings of the Advances in Neural Information Processing Systems*; Solla, S., Leen, T., Müller, K., Eds.; MIT Press: Cambridge, MA, USA, 1999; Volume 12.

66. Lohrasbinasab, I.; Shahraki, A.; Taherkordi, A.; Delia Jurcut, A. From statistical-to machine learning-based network traffic prediction. *Trans. Emerg. Telecommun. Technol.* **2022**, *33*, e4394. [[CrossRef](#)]
67. Ye, J.; Zhao, J.; Ye, K.; Xu, C. How to build a graph-based deep learning architecture in traffic domain: A survey. *IEEE Trans. Intell. Transp. Syst.* **2020**, *23*, 3904–3924. [[CrossRef](#)]
68. Wang, X.; Zhou, Z.; Xiao, F.; Xing, K.; Yang, Z.; Liu, Y.; Peng, C. Spatio-temporal analysis and prediction of cellular traffic in metropolis. *IEEE Trans. Mob. Comput.* **2018**, *18*, 2190–2202. [[CrossRef](#)]
69. He, K.; Huang, Y.; Chen, X.; Zhou, Z.; Yu, S. Graph attention spatial-temporal network for deep learning based mobile traffic prediction. In Proceedings of the 2019 IEEE Global Communications Conference (GLOBECOM), Waikoloa, HI, USA, 9–13 December 2019; pp. 1–6.
70. Yang, L.; Jiang, X.; Ji, Y.; Wang, H.; Abraham, A.; Liu, H. Gated graph convolutional network based on spatio-temporal semi-variogram for link prediction in dynamic complex network. *Neurocomputing* **2022**, *505*, 289–303. [[CrossRef](#)]
71. Kalander, M.; Zhou, M.; Zhang, C.; Yi, H.; Pan, L. Spatio-temporal hybrid graph convolutional network for traffic forecasting in telecommunication networks. *arXiv* **2020**, arXiv:2009.09849.
72. Zhou, X.; Zhang, Y.; Li, Z.; Wang, X.; Zhao, J.; Zhang, Z. Large-scale cellular traffic prediction based on graph convolutional networks with transfer learning. *Neural Comput. Appl.* **2022**, *34*, 5549–5559. [[CrossRef](#)]
73. Zhao, S.; Jiang, X.; Jacobson, G.; Jana, R.; Hsu, W.L.; Rustamov, R.; Talasila, M.; Aftab, S.A.; Chen, Y.; Borcea, C. Cellular network traffic prediction incorporating handover: A graph convolutional approach. In Proceedings of the 2020 17th Annual IEEE International Conference on Sensing, Communication, and Networking (SECON), Como, Italy, 22–25 June 2020; pp. 1–9.
74. Wei, G. A Summary of Traffic Flow Forecasting Methods. *J. Highw. Transp. Res. Dev.* **2004**, *21*, 82–85.
75. Feng, H.; Shu, Y. Study on network traffic prediction techniques. In Proceedings of the 2005 International Conference on Wireless Communications, Networking and Mobile Computing, Wuhan, China, 26 September 2005; Volume 2, pp. 1041–1044.
76. Dalgkitsis, A.; Louta, M.D.; Karetos, G.T. Traffic forecasting in cellular networks using the LSTM RNN. In Proceedings of the 22nd Pan-Hellenic Conference on Informatics, Athens, Greece, 29 November–1 December 2018.
77. Yu, L.; Li, M.; Jin, W.; Guo, Y.; Wang, Q.; Yan, F.; Li, P. Step: A spatio-temporal fine-granular user traffic prediction system for cellular networks. *IEEE Trans. Mob. Comput.* **2020**, *20*, 3453–3466. [[CrossRef](#)]
78. Wang, Z.; Hu, J.; Min, G.; Zhao, Z.; Chang, Z.; Wang, Z. Spatial-Temporal Cellular Traffic Prediction for 5 G and Beyond: A Graph Neural Networks-Based Approach. *IEEE Trans. Ind. Inform.* **2022**, *19*, 5722–5731. [[CrossRef](#)]
79. Berndt, D.J.; Clifford, J. Using Dynamic Time Warping to Find Patterns in Time Series. In Proceedings of the KDD Workshop, Newport Beach, CA, USA, 14–17 April 1994.
80. Lee, S.W.; sidgi Mohammed, H.; Mohammadi, M.; Rashidi, S.; Rahmani, A.M.; Masdari, M.; Hosseinzadeh, M. Towards secure intrusion detection systems using deep learning techniques: Comprehensive analysis and review. *J. Netw. Comput. Appl.* **2021**, *187*, 103111. [[CrossRef](#)]
81. Zhou, J.; Xu, Z.; Rush, A.M.; Yu, M. Automating botnet detection with graph neural networks. *arXiv* **2020**, arXiv:2003.06344.
82. Carpenter, J.; Layne, J.; Serra, E.; Cuzzocrea, A. Detecting Botnet Nodes via Structural Node Representation Learning. In Proceedings of the 2021 IEEE International Conference on Big Data (Big Data), Orlando, FL, USA, 15–18 December 2021; pp. 5357–5364.
83. Pujol-Perich, D.; Suarez-Varela, J.; Cabellos-Aparicio, A.; Barlet-Ros, P. Unveiling the potential of graph neural networks for robust intrusion detection. *ACM SIGMETRICS Perform. Eval. Rev.* **2022**, *49*, 111–117. [[CrossRef](#)]
84. Proterogerou, A.; Papadopoulos, S.; Drosou, A.; Tzovaras, D.; Refanidis, I. A graph neural network method for distributed anomaly detection in IoT. *Evol. Syst.* **2021**, *12*, 19–36. [[CrossRef](#)]
85. Lo, W.W.; Layeghy, S.; Sarhan, M.; Gallagher, M.; Portmann, M. E-graphsage: A graph neural network based intrusion detection system for iot. In Proceedings of the NOMS 2022—2022 IEEE/IFIP Network Operations and Management Symposium, Budapest, Hungary, 25–29 April 2022; pp. 1–9.
86. Chang, L.; Branco, P. Graph-based solutions with residuals for intrusion detection: The modified e-graphsage and e-resgat algorithms. *arXiv* **2021**, arXiv:2111.13597.
87. Deng, A.; Hooi, B. Graph neural network-based anomaly detection in multivariate time series. In Proceedings of the AAAI Conference on Artificial Intelligence, Virtual, 4–7 February 2021; Volume 35, pp. 4027–4035.
88. Zhao, J.; Liu, X.; Yan, Q.; Li, B.; Shao, M.; Peng, H. Multi-attributed heterogeneous graph convolutional network for bot detection. *Inf. Sci.* **2020**, *537*, 380–393. [[CrossRef](#)]
89. Sun, X.; Yang, J. HetGLM: Lateral Movement Detection by Discovering Anomalous Links with Heterogeneous Graph Neural Network. In Proceedings of the 2022 IEEE International Performance, Computing, and Communications Conference (IPCCC), Austin, TX, USA, 11–13 November 2022; pp. 404–411.
90. Lo, W.W.; Kulatilleke, G.; Sarhan, M.; Layeghy, S.; Portmann, M. XG-BoT: An explainable deep graph neural network for botnet detection and forensics. *Internet Things* **2023**, *22*, 100747. [[CrossRef](#)]
91. Zhu, X.; Zhang, Y.; Zhang, Z.; Guo, D.; Li, Q.; Li, Z. Interpretability evaluation of botnet detection model based on graph neural network. In Proceedings of the IEEE INFOCOM 2022—IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS), Austin, TX, USA, 11–13 November 2022; pp. 1–6.
92. Aouedi, O.; Piamrat, K.; Muller, G.; Singh, K. Federated semisupervised learning for attack detection in industrial Internet of Things. *IEEE Trans. Ind. Inform.* **2022**, *19*, 286–295. [[CrossRef](#)]

93. Ying, Z.; Bourgeois, D.; You, J.; Zitnik, M.; Leskovec, J. Gnnexplainer: Generating explanations for graph neural networks. *Adv. Neural Inf. Process. Syst.* **2019**, *32*, 9244–9255.
94. Yan, Z.; Ge, J.; Wu, Y.; Li, L.; Li, T. Automatic Virtual Network Embedding: A Deep Reinforcement Learning Approach with Graph Convolutional Networks. *IEEE J. Sel. Areas Commun.* **2020**, *38*, 1040–1057. [[CrossRef](#)]
95. Sun, P.; Lan, J.; Li, J.; Guo, Z.; Hu, Y. Combining Deep Reinforcement Learning with Graph Neural Networks for Optimal VNF Placement. *IEEE Commun. Lett.* **2021**, *25*, 176–180. [[CrossRef](#)]
96. Sun, P.; Lan, J.; Guo, Z.; Zhang, D.; Chen, X.; Hu, Y.; Liu, Z. DeepMigration: Flow Migration for NFV with Graph-based Deep Reinforcement Learning. In Proceedings of the ICC 2020—2020 IEEE International Conference on Communications (ICC), Dublin, Ireland, 7–11 June 2020; pp. 1–6. [[CrossRef](#)]
97. Habibi, F.; Dolati, M.; Khonsari, A.; Ghaderi, M. Accelerating Virtual Network Embedding with Graph Neural Networks. In Proceedings of the 2020 16th International Conference on Network and Service Management (CNSM), Izmir, Turkey, 2–6 November 2020; pp. 1–9. [[CrossRef](#)]
98. Zhang, P.; Wang, C.; Kumar, N.; Zhang, W.; Liu, L. Dynamic Virtual Network Embedding Algorithm Based on Graph Convolution Neural Network and Reinforcement Learning. *IEEE Internet Things J.* **2022**, *9*, 9389–9398. [[CrossRef](#)]
99. Heo, D.; Lange, S.; Kim, H.G.; Choi, H. Graph Neural Network based Service Function Chaining for Automatic Network Control. In Proceedings of the 2020 21st Asia-Pacific Network Operations and Management Symposium (APNOMS), Daegu, Republic of Korea, 22–25 September 2020; pp. 7–12. [[CrossRef](#)]
100. Qi, S.; Li, S.; Lin, S.; Saidi, M.Y.; Chen, K. Energy-Efficient VNF Deployment for Graph-Structured SFC Based on Graph Neural Network and Constrained Deep Reinforcement Learning. In Proceedings of the 2021 22nd Asia-Pacific Network Operations and Management Symposium (APNOMS), Tainan, Taiwan, 8–10 September 2021; pp. 348–353. [[CrossRef](#)]
101. Pei, J.; Hong, P.; Li, D. Virtual Network Function Selection and Chaining Based on Deep Learning in SDN and NFV-Enabled Networks. In Proceedings of the 2018 IEEE International Conference on Communications Workshops (ICC Workshops), Kansas City, MO, USA, 20–24 May 2018; pp. 1–6. [[CrossRef](#)]
102. Hasselt, H.v.; Guez, A.; Silver, D. Deep Reinforcement Learning with Double Q-Learning. In Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence, AAAI’16, Phoenix, AZ, USA, 12–17 February 2016; AAAI Press: Washington, DC, USA, 2016; pp. 2094–2100.
103. Moy, J. OSPF Version 2. RFC 2328. 1998. Available online: <https://www.rfc-editor.org/info/rfc2328> (accessed on 1 October 2023).
104. Rekhter, Y.; Hares, S.; Li, T. A Border Gateway Protocol 4 (BGP-4). RFC 4271. 2006. Available online: <https://www.rfc-editor.org/info/rfc4271> (accessed on 1 October 2023).
105. Rusek, K.; Cholda, P. Message-Passing Neural Networks Learn Little’s Law. *IEEE Commun. Lett.* **2019**, *23*, 274–277. [[CrossRef](#)]
106. Rusek, K.; Suárez-Varela, J.; Mestres, A.; Barlet-Ros, P.; Cabellos-Aparicio, A. Unveiling the Potential of Graph Neural Networks for Network Modeling and Optimization in SDN. In Proceedings of the 2019 ACM Symposium on SDN Research, San Jose, CA, USA, 3–4 April 2019; pp. 140–151. [[CrossRef](#)]
107. Badia-Sampera, A.; Suárez-Varela, J.; Almasan, P.; Rusek, K.; Barlet-Ros, P.; Cabellos-Aparicio, A. Towards More Realistic Network Models Based on Graph Neural Networks. In Proceedings of the 15th International Conference on Emerging Networking Experiments and Technologies, Orlando, FL, USA, 9–12 December 2019; pp. 14–16. [[CrossRef](#)]
108. Suárez-Varela, J.; Carol-Bosch, S.; Rusek, K.; Almasan, P.; Arias, M.; Barlet-Ros, P.; Cabellos-Aparicio, A. Challenging the Generalization Capabilities of Graph Neural Networks for Network Modeling. In Proceedings of the ACM SIGCOMM 2019 Conference Posters and Demos, Beijing, China, 19–23 August 2019; pp. 114–115. [[CrossRef](#)]
109. Ferriol-Galmés, M.; Suárez-Varela, J.; Paillissé, J.; Shi, X.; Xiao, S.; Cheng, X.; Barlet-Ros, P.; Cabellos-Aparicio, A. Building a Digital Twin for Network Optimization Using Graph Neural Networks. *Comput. Netw.* **2022**, *217*, 109329. [[CrossRef](#)]
110. Ferriol-Galmés, M.; Paillisse, J.; Suárez-Varela, J.; Rusek, K.; Xiao, S.; Shi, X.; Cheng, X.; Barlet-Ros, P.; Cabellos-Aparicio, A. RouteNet-Fermi: Network Modeling with Graph Neural Networks. *arXiv* **2022**, arXiv:2212.12070.
111. Sawada, K.; Kotani, D.; Okabe, Y. Network Routing Optimization Based on Machine Learning Using Graph Networks Robust against Topology Change. In Proceedings of the 2020 International Conference on Information Networking (ICOIN), Barcelona, Spain, 7–10 January 2020; pp. 608–615. [[CrossRef](#)]
112. Chen, B.; Zhu, D.; Wang, Y.; Zhang, P. An Approach to Combine the Power of Deep Reinforcement Learning with a Graph Neural Network for Routing Optimization. *Electronics* **2022**, *11*, 368. [[CrossRef](#)]
113. Sun, P.; Guo, Z.; Lan, J.; Li, J.; Hu, Y.; Baker, T. ScaleDRL: A Scalable Deep Reinforcement Learning Approach for Traffic Engineering in SDN with Pinning Control. *Comput. Netw.* **2021**, *190*, 107891. [[CrossRef](#)]
114. Xu, Z.; Tang, J.; Meng, J.; Zhang, W.; Wang, Y.; Liu, C.H.; Yang, D. Experience-Driven Networking: A Deep Reinforcement Learning Based Approach. In Proceedings of the IEEE INFOCOM 2018—IEEE Conference on Computer Communications, Honolulu, HI, USA, 16–19 April 2018; pp. 1871–1879. [[CrossRef](#)]
115. Swaminathan, A.; Chaba, M.; Sharma, D.K.; Ghosh, U. GraphNET: Graph Neural Networks for Routing Optimization in Software Defined Networks. *Comput. Commun.* **2021**, *178*, 169–182. [[CrossRef](#)]
116. Boyan, J.; Littman, M. Packet Routing in Dynamically Changing Networks: A Reinforcement Learning Approach. *Adv. Neural Inf. Process. Syst.* **1993**, *6*, 671–678.
117. Huang, R.; Guan, W.; Zhai, G.; He, J.; Chu, X. Deep Graph Reinforcement Learning Based Intelligent Traffic Routing Control for Software-Defined Wireless Sensor Networks. *Appl. Sci.* **2022**, *12*, 1951. [[CrossRef](#)]

118. Stampa, G.; Arias, M.; Sanchez-Charles, D.; Muntés-Mulero, V.; Cabellos, A. A Deep-Reinforcement Learning Approach for Software-Defined Networking Routing Optimization. *arXiv* **2017**, arXiv:1709.07080.
119. Varga, A. The OMNeT++ Discrete Event Simulation System. In Proceedings of the European Simulation Multiconference (ESM'2001), Prague, Czech Republic, 6–9 June 2001.
120. Williams, R.J. Simple Statistical Gradient-Following Algorithms for Connectionist Reinforcement Learning. *Mach. Learn.* **1992**, *8*, 229–256. [[CrossRef](#)]
121. Konda, V.; Tsitsiklis, J. Actor-Critic Algorithms. *Adv. Neural Inf. Process. Syst.* **1999**, *12*, 1008–1014.
122. Wang, H.; Wu, Y.; Min, G.; Miao, W. A Graph Neural Network-Based Digital Twin for Network Slicing Management. *IEEE Trans. Ind. Inform.* **2022**, *18*, 1367–1376. [[CrossRef](#)]
123. Xu, Y.; Zhang, Y.; Guo, W.; Guo, H.; Tang, R.; Coates, M. Graphsail: Graph structure aware incremental learning for recommender systems. In Proceedings of the 29th ACM International Conference on Information & Knowledge Management, Virtual, 19–23 October 2020; pp. 2861–2868.
124. Huang, V.; Sohail, S.; Mayo, M.; Botran, T.L.; Rodrigues, M.; Anderson, C.; Ooi, M. Keep It Simple: Fault Tolerance Evaluation of Federated Learning with Unreliable Clients. *arXiv* **2023**, arXiv:2305.09856.
125. Xu, C.; Qu, Y.; Xiang, Y.; Gao, L. Asynchronous federated learning on heterogeneous devices: A survey. *arXiv* **2021**, arXiv:2109.04269.
126. Zhang, X.; Zitnik, M. GnnGuard: Defending graph neural networks against adversarial attacks. *Adv. Neural Inf. Process. Syst.* **2020**, *33*, 9263–9275.
127. Zhu, D.; Zhang, Z.; Cui, P.; Zhu, W. Robust graph convolutional networks against adversarial attacks. In Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, Anchorage, AK, USA, 4–8 August 2019; pp. 1399–1407.
128. Zügner, D.; Akbarnejad, A.; Günnemann, S. Adversarial attacks on neural networks for graph data. In Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, London, UK, 19–23 August 2018; pp. 2847–2856.
129. He, C.; Balasubramanian, K.; Ceyani, E.; Yang, C.; Xie, H.; Sun, L.; He, L.; Yang, L.; Yu, P.S.; Rong, Y.; et al. FedGraphNN: A Federated Learning System and Benchmark for Graph Neural Networks. *arXiv* **2021**, arXiv:2104.07145.
130. Zhang, K.; Yang, C.; Li, X.; Sun, L.; Yiu, S.M. Subgraph Federated Learning with Missing Neighbor Generation. *Adv. Neural Inf. Process. Syst.* **2021**, *34*, 6671–6682.
131. Liu, R.; Xing, P.; Deng, Z.; Li, A.; Guan, C.; Yu, H. Federated Graph Neural Networks: Overview, Techniques and Challenges. *arXiv* **2022**, arXiv:2202.07256.
132. Castro, F.M.; Marín-Jiménez, M.J.; Mata, N.G.; Schmid, C.; Karteek, A. End-to-End Incremental Learning. In Proceedings of the European Conference on Computer Vision, Munich, Germany, 8–14 September 2018.
133. Kirkpatrick, J.; Pascanu, R.; Rabinowitz, N.C.; Veness, J.; Desjardins, G.; Rusu, A.A.; Milan, K.; Quan, J.; Ramalho, T.; Grabska-Barwinska, A.; et al. Overcoming catastrophic forgetting in neural networks. *Proc. Natl. Acad. Sci. USA* **2016**, *114*, 3521–3526. [[CrossRef](#)] [[PubMed](#)]
134. McMahan, B.; Moore, E.; Ramage, D.; Hampson, S.; Arcas, B.A.y. Communication-Efficient Learning of Deep Networks from Decentralized Data. In Proceedings of the 20th International Conference on Artificial Intelligence and Statistics, PMLR, Fort Lauderdale, FL, USA, 20–22 April 2017; Volume 54, pp. 1273–1282.
135. Pei, Y.; Mao, R.; Liu, Y.; Chen, C.; Xu, S.; Qiang, F. Decentralized Federated Graph Neural Networks. In Proceedings of the International Workshop on Federated and Transfer Learning for Data Sparsity and Confidentiality in Conjunction with IJCAI 2021, Montreal, QC, Canada, 21 August 2021.
136. Wu, C.; Wu, F.; Cao, Y.; Huang, Y.; Xie, X. FedGNN: Federated Graph Neural Network for Privacy-Preserving Recommendation. *Nat. Commun.* **2022**, *13*, 3091. [[CrossRef](#)] [[PubMed](#)]
137. Fu, X.; Zhang, J.; Meng, Z.; King, I. Magnn: Metapath aggregated graph neural network for heterogeneous graph embedding. In Proceedings of the Web Conference 2020, Taipei, Taiwan, 20–24 April 2020; pp. 2331–2341.
138. Jiang, X.; Lu, Y.; Fang, Y.; Shi, C. Contrastive pre-training of gnns on heterogeneous graphs. In Proceedings of the 30th ACM International Conference on Information & Knowledge Management, Virtual, 1–5 November 2021; pp. 803–812.
139. Sun, Y.; Wang, S.; Tang, X.; Hsieh, T.Y.; Honavar, V. Adversarial attacks on graph neural networks via node injections: A hierarchical reinforcement learning approach. In Proceedings of the Web Conference 2020, Taipei, Taiwan, 20–24 April 2020; pp. 673–683.
140. Ferriol-Galmes, M.; Rusek, K.; Suarez-Varela, J.; Xiao, S.; Shi, X.; Cheng, X.; Wu, B.; Barlet-Ros, P.; Cabellos-Aparicio, A. RouteNet-Erlang: A Graph Neural Network for Network Performance Evaluation. In Proceedings of the IEEE INFOCOM 2022—IEEE Conference on Computer Communications, London, UK, 2–5 May 2022; pp. 2018–2027. [[CrossRef](#)]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.