Graph Neural Networks in Computer Networks: A Survey

Ling Xia Liao[1], Yulan Xu[2], Xuxia Liang[3], Han-Chieh Chao[4,5], Shubao Pan[1]*

[1] School of Artifitual Intelligence, Guilin University of Aerospace Technology, China
[2] School of Electronic Engineering and Automation, Guilin University of Electronic Technology, China
[3] Department of Obstetrics, the People's Hospital of Guangxi Zhuang Autonomous Region, Guangxi Academy of Medical Sciences, China
[4] Department of Applied Informatics, Fo Guang University, Taiwan
[5] Department of Electrical Engineering, National Dong Hwa University, Taiwan
liaolx@guat.edu.cn, xuyulan@mails.guet.edu.cn, 1345067634@qq.com, hcc@fgu.edu.tw, panshubao@guat.edu.cn

Abstract

Traditional methods of analyzing and processing computer networks have difficulty dealing with their rapidly increasing scalability, structure, and traffic complexity due to the rapid development of information technology. Graph neural networks (GNNs), an emerging deep learning technology, adapt to graph-structured data and relationships. They offer new mechanisms and solutions to real-world problems. This paper provides a structured, unified synthesis of current research in this interdisciplinary field. It reviews the basic concepts and models of GNNs, highlights their current state in major computer network management and optimization scenarios, and discusses typical applications, challenges, open issues, and future research directions.

Keywords: Graph neural network, Computer networks, Deep learning, Internet of Things, Industrial Internet of Things

1 Introduction

Graph Neural Networks (GNNs) are neural network models for graph-structured data. They learn graph structure features and capture hidden patterns in these structures by aggregating neighborhood information. First introduced in 2008 [1], GNNs have rapidly been applied to complex interaction scenarios such as social networks, recommendation systems, and transportation networks [2], which have complex, non-IDD (independently distributed), or cross-domain data. Traditional machine learning methods struggle to address these scenarios [3].

Although computer networks enable data communication, social networks, computer vision, and recommender systems deal with human interactions, visual data, and personalized ranking, respectively, all of these types of networks have graph-structured data. Currently, managing and optimizing computer networks is challenging. For example, traditional iterative update mechanisms struggle to update network topology quickly in large-scale, dynamic scenarios with millisecond responses, such as 6G network slicing [4]; classical Federated Learning (FL) methods lack mechanisms to protect network privacy and security without consuming excessive network bandwidth and computational resources [5]; accurate detection of hierarchical attacks is difficult due to the lack of a multi-server trust system and universal verification of adversarial attack defense [6]; and existing multimodal fusion lacks interpretability, relies too heavily on attention-weighted features, and does not consider heterogeneous data causality [7].

GNNs are better suited for computer network management and optimization than traditional methods such as table features and matrix features. GNNs can naturally model computer networks. A computer network can be viewed as an attributed dynamic graph with dynamic properties. In these model, hosts, switches, routers, and firewalls are nodes, and traffic paths and links are edges. Edge attributes can include bandwidth, latency, packet loss rate, and link stability, while node attributes can include CPU load, queue length, storage capacity, and fault status. Therefore, GNNs can use neighbor aggregation and multi-hop propagation to directly capture local and global topological features and dynamic evolution, which are difficult to fully express using traditional methods. GNNs can automatically learn the multidimensional relationships between nodes, edges, and traffic without manual feature design. As the network scale increases, the operating environment migrates, or the attack model is updated, traditional methods require retraining of the models to adapt to new knowledge. In contrast, GNNs can adapt models to the new nodes and edges, and new patterns and knowledge behind the changes through inductive learning, parameter adjustment, and self-supervised/contrastive learning. Dynamic GNNs can naturally model time series changes and achieve continuous reasoning. Depending on the downstream tasks, network traffic or traffic forwarding rules can also construct graphs for GNNs.

GNNs have been applied in computer networks to improve dynamic graph modeling [8], real-time task scheduling, energy efficiency optimization [9], and the integration of homomorphic encryption with federated learning and other machine learning algorithms. However, they have not been systematically discussed, analyzed, or compared for advanced use in many other network scenarios [10].

*Corresponding Author: Shubao Pan; Email: panshubao@guat.edu.cn
DOI: https://doi.org/10.70003/160792642026012701010

A review of GNN scenarios and applications in computer networks is necessary to make breakthroughs towards time-efficient, dynamic, scalable, and secure GNN mechanisms.

This paper reviews the basic concepts, classical models, usage scenarios, and applications of GNNs in computer networks. The goal is threefold: to consolidate the rapidly growing yet fragmented field of GNN research; to emphasize the unique value that GNNs offer in solving network problems; and to guide future research, deployment, and standardization efforts. This paper can serve as a foundational resource for networking and machine learning research. To the best of our knowledge, this study is the first to directly link GNNs to related scenarios and applications in computer networks. It makes the following main contributions.

(1) Six GNN models are discussed from multiple dimensions.

(2) The six major use cases are categorized, including topology management, routing optimization, traffic forecasting, cybersecurity, and resource management. These use cases extend to the Internet of Things (IoT) and the Industrial Internet of Things (IIoT). Typical GNN applications in each scenario are reviewed to explain how and why GNNs should be involved.

(3) The challenges and open issues are identified. Future research directions for emerging networking scenarios and new technologies are suggested.

The rest of this paper is organized as follows: Section 2 provides related surveys. Sections 3 and 4 offer brief introductions to the fundamental concepts and models of GNNs, respectively. Section 5 discusses GNN scenarios and applications in computer networks. Section 6 addresses the major challenges and open issues of GNNs. Section 7 concludes the paper.

2 Related Surveys

As listed in Table 1, Shabani et al. [11] and Khemani et al. [12] provided thorough reviews of GNN models, algorithms, and methods. Vatter et al. [13] discussed the structure of distributed GNNs; Wu et al. [14] focused on GNN computational architecture; and Oloulade et al. classified, searched, and optimized GNN models [15]. Liu et al. [16] and Munikoti et al. [17] discussed GNNs in the context of Federated Learning (FL) and Reinforcement Learning (RL), respectively. Nandan et al. [18], Wang et al. [19], and Liu et al. [20] investigated the GNNs' representability, uncertainty, and tasks. While Wu et al. surveyed GNN usage scenarios and applications in recommendation systems [21], Dong et al. surveyed them in the IoTs [22], and Rahmani et al., Chen et al., and Wu et al. summarized them in transportation systems, computer vision, and natural language processing, respectively [23-25], to the best of our knowledge, no existing surveys focus specifically on GNN applications in computer networks. This study fills this gap.

Table 1. Related surveys

Survey perspective		References	Publication year
Comprehensive		[11-12]	2024
Technology	Distributed GNN	[13]	2023
	Computer architecture	[14]	2025
	Model search	[15]	2022
	FL	[16]	2025
	RL	[17]	2024
	Explainability	[18]	2025
	Uncertainty	[19]	2024
	Type of task	[20]	2023
Application	Recommender systems	[21]	2022
	IoTs	[22]	2023
	Intelligent transportation	[23]	2023
	Computer vision	[24]	2024
	Natural language process	[25]	2023

3 Fundamental of GNNs

Graph-structured data can be denoted by $G = (V, E)$, where V and E are sets of vertices and edges, representing the entities and their relationships, respectively [26]. Unlike images, texts, and time series data, which are Euclidean structures with a fixed arrangement and order of nodes, graphs are non-Euclidean structures. Since neural networks are designed for Euclidean data, they are difficult to apply directly to graphs. GNNs can learn vertex, edge, and graph representations for vertex classification and edge prediction tasks [27].

A GNN typically consists of an input module, a message-passing module, and an aggregation and update module. Figure 1(a) shows a k-layer GNN. The input module is located in layer 0, where the graph structure data is provided and the embeddings are initialized. In each subsequent layer, the nodes collect information from their neighbors through the message-passing module and update their embeddings through the aggregation and update module. This forms an iterative process called message passing [28]. After k rounds of message passing and feature updating, each node learns a representation containing structural information about itself and its neighbors. Figure 1(b) illustrates this process. Here, h_v^k represents the kth layer embedding of node v. This process is the core of GNNs. It simulates the topological relationships of graph structure data and successfully overcomes the bottleneck that traditional neural networks experience when processing non-Euclidean data.

4 GNN Models

4.1 Graph Convolutional Networks

Graph Convolutional Networks (GCNs) extract local features by convolving the features of neighboring nodes. A weighted adjacency matrix defines the contribution of neighboring nodes to improve performance and stability. GCNs can efficiently process graph structure data for node and graph classification tasks. A GCN usually consists of graph convolution layers, nonlinear activation functionl ayers, and fully connected layers, which aggregate information from neighboring nodes, introducing nonlinear factors, and outputting the final data results, respectively [29]. Both GCN and Convolutional Neural Network (CNN) rely on convolutional structure to process data, extract features from neighboring nodes, and increase the perceptual scale for the broad tasks using a multi-layer structure, as shown in Figure 2. However, CNNs deal with grid-based images and audio in the European space, while GCNs deal with graph structures in the non-European space. Additionally, CNNs have a fixed neighborhood in which the convolution kernel slides, whereas GCNs have a flexible neighborhood defined by graph topology in which information is aggregated.

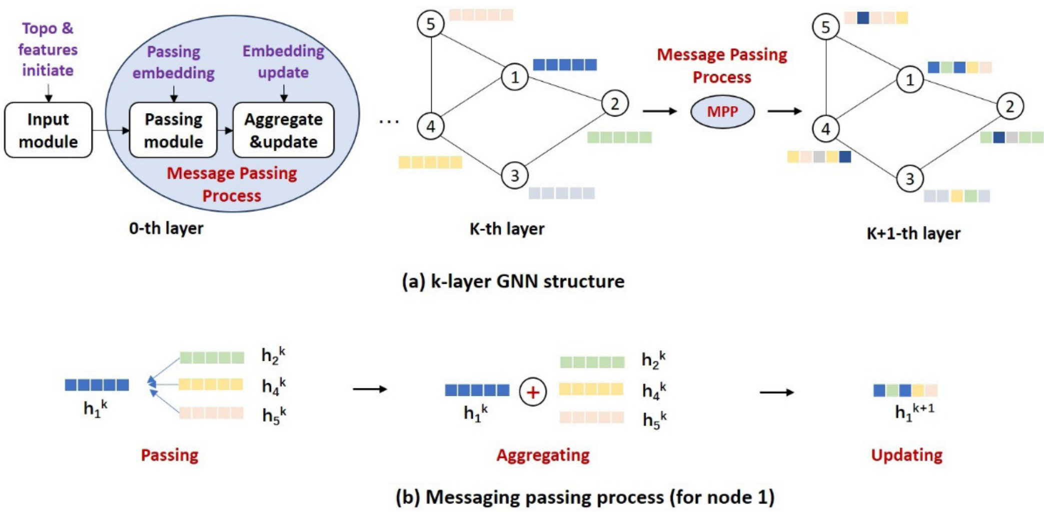

Figure 1. The high-level structure of a GNN and the message passing process

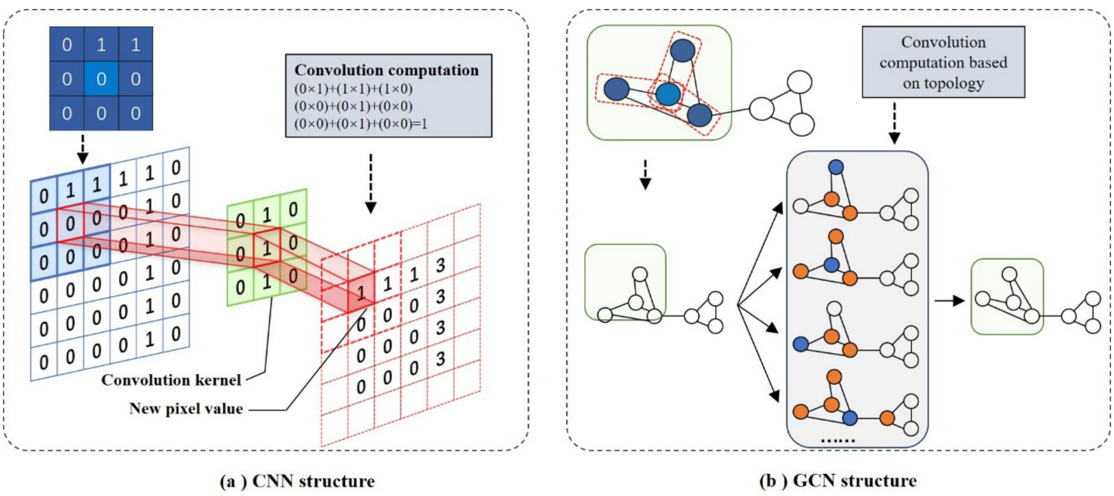

Figure 2. The structures of CNN and GCN

4.2 Graph Attention Transformers

GCNs rely on a full graph Fourier transform to aggregate node features, which prevents them from scaling to large dynamic graphs. This leads to the development of Graph Attention Transformers (GATs). GATs use attention mechanisms to assign different weights to neighboring nodes and adaptively aggregate neighboring features for heterogeneous graph structure data. By focusing on more relevant neighbor nodes, GATs are more flexible and effective when dealing with complex graph structure data.

Figure 3(a) illustrates an attention mechanism. Here h_i and h_j represent the features of nodes i and j, respectively. $\vec{Wh}_i$ and $\vec{Wh}_j$ are their initial weights. $\vec{a}$ is a learnable matrix, and α_{ij} is an element of $\vec{a}$ that represents the weight from h_i to h_j. It can be calculated by Eq. (1), where $LeakyReLU$ is the activation function, T is a transformer operation, and || is the concatenation operation. Figure 3(b) illustrates a GAT's structure, where node 1's features are updated by aggregating the features of nodes 2, 3, and 4 with the weights computed by one-head attention.

$$\alpha_{ij} = \frac{\exp\left(LeakyReLU\left(a^T\left[Wh_i \,\|\, Wh_j\right]\right)\right)}{\sum_{k\in N(i)}\exp\left(LeakyReLU\left(a^T\left[Wh_i \,\|\, Wh_k\right]\right)\right)} \tag{1}$$

GATs can capture fine-grained relationships in graphs and have demonstrated superior performance in tasks such as node classification and link prediction. This design overcomes the limitation of symmetric adjacency matrices in GCNs [30], and the multi-head attention mechanism improves the model's ability to express heterogeneous relationships by computing multiple independent attention sets in parallel.

4.3 GraphSAGE

GraphSAGE introduces neighborhood sampling and hierarchical aggregation strategies to address the limitations of GCNs in processing industrial-scale massive graphs. As shown in Figure 4, GraphSAGE first samples a neighborhood, then aggregates feature information from the sampled neighbors, and finally processes a specific task based on the aggregated information. GraphSAGE can use mini-batch training to integrate randomly sampled neighborhood information layer by layer through learnable aggregation functions, such as long short-term memory (LSTM) and pooling. This approach alleviates computational memory pressure and is effective when dealing with dynamic or constantly expanding graphs. It also achieves zero-shot inference of new nodes and has been successfully applied to model billion-level community networks, such as the Ali e-commerce platform.

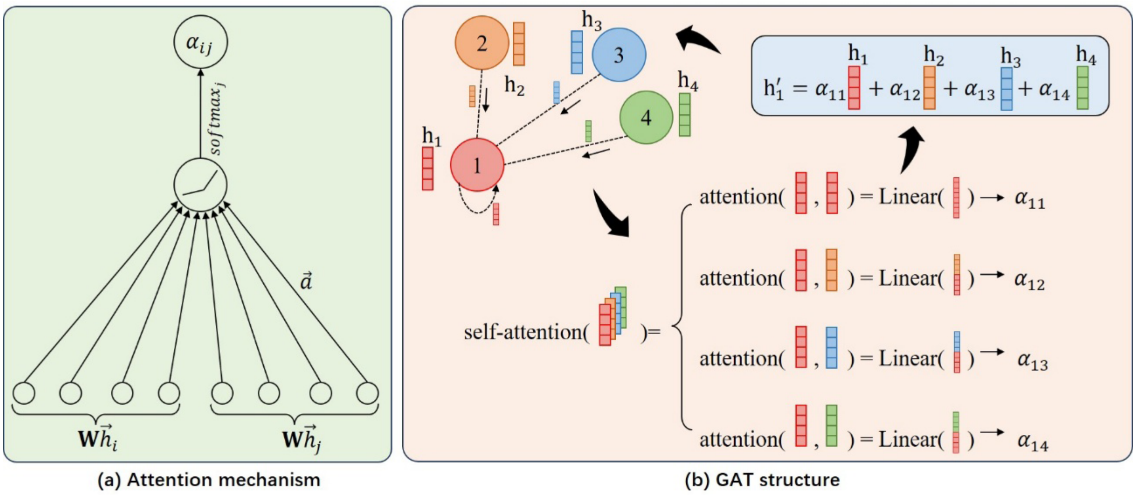

Figure 3. Attention mechanism and the GAT structure

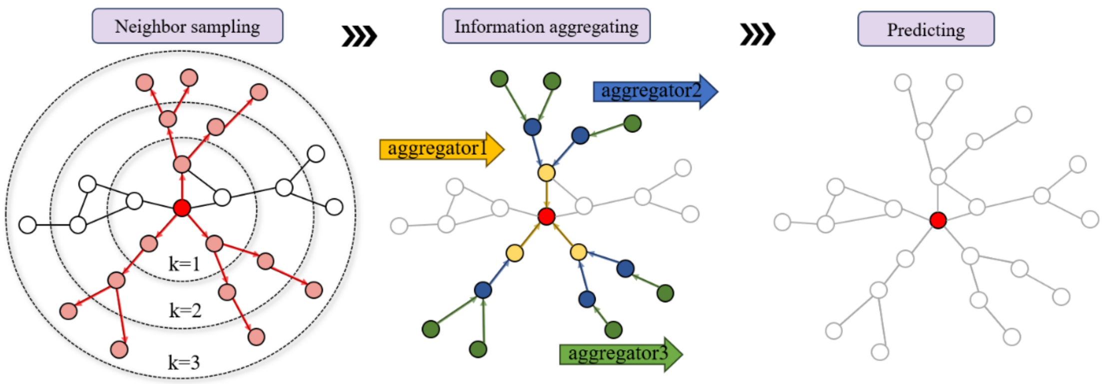

Figure 4. The high-level structure of GraphSAGE

4.4 Graph Recurrent Networks

Graph Recurrent Networks (GRNs) [31] target dynamic graph data. GRNs model time series dependence and node state evolution by combining graph structure and Recurrent Neural Networks (RNNs). GRNs integrate graph recurrent unit or LSTM through a gated mechanism to model spatio-temporal correlations among node features.

RNNs have a recurrent structure that enables them to recall previous information to improve predictions about subsequent elements in a sequence. Figure 5(a) shows a GNN with a hidden state Z_t, which is updated using the perceptron $Z_t = \sigma(Ax_t + Bz_{t-1})$. It also has an output prediction $y_t = \sigma(Cx_t)$. Figure 5(b) illustrates how a GNN is incorporated into a GRN to address dynamic information and time series data in graphs. Typical GRN research includes dynamic graph representation learning, temporal graph generation, and multi-way prediction optimization.

4.5 Summary

As shown in Table 2, GCN employs graph convolution to average neighbor features for small-scale graph tasks with simple structures and balanced node relationships. GAT uses attention mechanisms to adaptively learn the importance of neighboring nodes. It is suitable for graphs with unbalanced adjacency relationships or complex structures. GraphSAGE incorporates sampling and aggregation to efficiently train on large graphs. GRN uses RNNs to propagate states on graphs, modeling long-term dependencies and temporal evolution between nodes in dynamic graphs or time-critical scenarios. GCN and GAT are designed for entire graphs, whereas GraphSAGE and GRN are better suited for modeling large or dynamic graphs.

Other GNNs, such as the Graph AutoEncoder (GAE) [32] and Graph-BERT [33] models, are also widely studied in current research. All of these models support supervised, semi-supervised, and unsupervised learning. However, GAE and Graph-BERT are primarily designed for unsupervised learning, and GCN is mainly designed for semi-supervised learning.

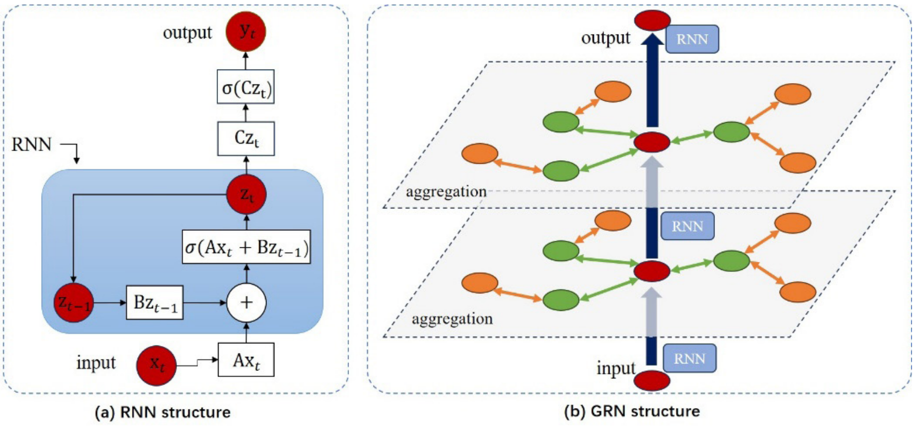

Figure 5. The high-level structure of a RNN and a GRN with the RNN

Table 2. Comparison of typical GNN models

Dimensions	GCN	GAT	GraphSAGE	GRN
Aggregation	Mean/Laplace convolution	Attention weighted aggregation	Mean/LSTM/Pooling	Based on RNNs
Neighbor weights	Fixed (normalized based on degree)	weights may not be learnable	Dynamic	determined by RNN hidden states
Information propagation	Fixed inter-layer propagation structure	Adaptive attention to neighbor importance	Local sampling + aggregation	Multi-step propagation + state update
For large graph	Not suitable	Not suitable	suitable	suitable
Expression	Moderate	Stronger	Controllable	Strong
Over smoothing	Prone to over-smoothing	Mild (weight adjustment)	Mitigated (reduce redundant sampling)	Alleviated (state retains history)
Parameter count	Low	High (attention head)	Depends on aggregation function	High (RNN parameters)
Tasks	Traffic classification Routing optimization Anomaly detection	Topo manag. Resource manag. cybersecurity	Topo&traffic forecast Intrusion detection Dynamic nodes	Traffic forecast Anomaly detection Routing optimization

5 Scenarios and Applications of GNNs in Computer Networks

5.1 Topology Management

Topology management involves understanding, modeling, and optimizing the interconnection structure of devices, such as switches and routers, in a computer network. The goal is to enable visualization, prediction, planning, and management of the network. Traditional heuristic algorithms, such as OSPF, are designed for networks with static topologies. Therefore, they have difficulty finding the optimal path in today's networks, which have time-varying topologies caused by 5G network slicing and the IoTs and IIoTs. Efficient topology discovery is essential in these networks to enable a flexible routing strategy for predicting network traffic and balancing loads. GNNs can directly model device connectivity and aggregate multi-hop neighbor information, they can also extract graph features such as degree and path count in an automated, embeddable, and learnable manner. GNNs can train models on small graphs and historical data and then generalize to large graphs and future scenarios. By combining RNN and Transformer modules, GNNs have been widely applied to enable dynamic topology management in today's networks.

In [34], topological data analysis (TDA) was proposed to enhance the representation of topology with rings and holes by incorporating GNNs. This method involves topology-sensitive graph representation learning. In [35], KerGNNs were introduced by combining graph kernels and GNNs to capture complex node interactions effectively and improve traditional GNN representations. However, KerGNNs still face challenges regarding computational complexity and parameter tuning for fully connected graphs and when multiple hyperparameters are involved.

5.2 Routing Optimization

Routing optimization involves determining the most efficient path for data to travel over a computer network. The goal is to minimize latency, maximize throughput, balance loads, and meet quality-of-service constraints. However, dynamic traffic patterns, changing topologies, multi-objective trade-offs, global dependencies, and the need for increased network scalability pose challenges to routing optimization in today's networks. GNNs are well-suited for routing optimization because networks are graph-structured by nature and routing involves topology-aware decision-making.

Reference [36] proposed a routing algorithm based on GNN and Deep Reinforcement Learning (DRL) for SDNs. The GNN model was trained on the SDN controller to dynamically optimize the packet transmission path. The model is highly robust to changes in network topology, such as link addition or deletion.

Reference [37] optimized routing by integrating a GNN into a DRL agent and designing a specific action space for generalization. The DRL+GNN agent updated the link state during the message passing process. An RNN captured the link state changes, and a Deep Neural Network (DNN) was used to output Q-value estimates.

As described in reference [38], AutoGNN combines GNN and DRL to automatically generate routing decisions. GNNs address traffic distribution within network topologies, while DRL trains GNN parameters. AutoGNN showed robustness to topology changes.

Reference [39] summarizes the integration potential of GNNs and DRLs in end-to-end (E2E) networks. The combination of GNNs and DRLs shows great potential for optimizing network resources, routing, and management, as well as contactless automation.

5.3 Traffic Forecasting

A traffic forecast predicts the future traffic conditions, such as bandwidth utilization, packet volume, and flow dynamics, over time and across network links or nodes. An accurate forecast supports smarter decisions regarding routing, resource allocation, and congestion control. Traditional forecast models (e.g., ARIMA, LSTM, and CNN) focus on time series and ignore the spatial dependencies within the network. In real networks, however, traffic at one node or link directly affects its neighbors. GNNs best capture this spatial coupling, leading to accurate, topology-sensitive, and proactive predictions for modern network management and routing optimization.

Lin and Wang [40] proposed a multi-time scale prediction model trained by CNN-GRU to capture the rapid changes in traffic values over short time intervals and address the burstiness of fine-grained network traffic. Experiments showed that this approach outperforms the baseline and reduces the burst traffic prediction error in China Unicom's single-cell dataset.

Reference [42] introduced a time-series similarity-based graph attention network (TSGAN) to predict cellular traffic and allocate cellular network resources proactively and effectively. Simulations showed that the TSGAN outperformed three classical prediction models on a real cellular network dataset in short-, medium-, and long-term prediction scenarios.

Reference [43] models E2E delay by using a GNN to learn correlations between global and basic network behaviors. A packet-level load balancing scheme within programmable data planes was also proposed to balance data plane traffic. Experiments demonstrated the feasibility and effectiveness of these approaches. Compared to queueing theory, RouteNet, and GNN-based schemes, the proposed approach improved the goodness of fit (R^2) and generalization ability under unknown traffic control strategies.

5.4 Cybersecurity

Cybersecurity is essential to safeguarding network infrastructure, sensitive data, and essential services from cyber threats such as malware, phishing, denial-of-service attacks, and data breaches. Maintaining cybersecurity is essential to ensuring national security, business continuity, public safety, and individual privacy. GNNs are particularly effective in cybersecurity because many cyber systems and threats exhibit graph-like structures, such as host and network graphs, program behavior graphs, phishing and social attack graphs, and attack path graphs. Compared to current classical cybersecurity approaches, GNNs offer topology awareness, contextual understanding, scalability, generalization, and reduced false positives. These capabilities allow GNNs to capture the structural, relational, and temporal complexity of cyber environments, enabling them to detect and defend against modern cyber threats accurately, adaptively, and scalably.

5.4.1 Intrusion Detection

Intrusion Detection Systems (IDS) monitor and analyze network or system activity to detect unauthorized access, malicious behavior, and policy violations. Traditional IDS (especially in signature-based IDS) struggle to detect unknown attacks. They often produce high false positives in anomaly-based models and are ineffective at modeling complex, multi-stage, or distributed attacks. They also struggle to exploit structural or contextual relationships in traffic or behavior. GNNs are widely used because they can naturally model relationships between entities (e.g. hosts, packets, and processes) and to learn from structural context.

Lo et al. [7] proposed E-GraphSAGE, a GNN-based IDS for the IoT environment. GraphSAGE captures connection and interaction patterns between devices. It learns device behaviors to detect abnormal and intrusive activities.

Chang et al. [44] proposed the E-GraphSAGE and E-ResGAT algorithms for intrusion attack detection. Residual links were introduced into the graph to preserve original node information and increase identification sensitivity for certain malicious traffic categories, improving class imbalance.

Nguyen et al. [45] studied intrusion detection in microservice architectures and proposed the DeepTraLog model for microservice anomaly detection. DeepTraLog combines trace logs and graph-based deep learning to detect abnormal remote procedure call traffic in containerized microservices.

Regarding intrusion detection based on GNNs in IoT systems with limited budgets, Zhou et al. [6] proposed a new hierarchical adversarial attack generation method that implements a hierarchy-aware black-box adversarial attack strategy. Adversarial samples are generated by hierarchically selecting high-priority vulnerable nodes based on the shadow GNN model by combining saliency mapping and the random walk restart algorithm. This research reveals the vulnerability of existing GNN models in IoT security scenarios.

5.4.2 Malware Detection

Malware detection involves identifying malicious software, such as viruses, worms, ransomware, Trojans, and spyware, that can compromise the confidentiality, integrity, or availability of computer systems. The goal is to accurately and reliably classify whether a program, file, or process exhibits malicious behavior at an early stage, including for obfuscated or unknown threats. Traditional approaches typically rely on static signatures, which can be easily circumvented by code obfuscation or repackaging. These approaches also lack insight into the complex structural behavior of malware, resulting in high false positives and negatives. However, since malware often exhibits relational, structured, and graph-like behavior—for example, system/API calls form a behavior graph, binary codes form a control flow graph, function relationships form a call graph, and malware families share structural patterns in graphs—GNNs can effectively model these structures. Thus, GNNs can enable robust malware detection that outperforms traditional static or rule-based methods [46].

Busch et al. [47] proposed a method for detecting and classifying malware based on network flow graphs. First, a flow graph is constructed by dynamically analyzing network traffic during application execution to create a richer representation of network communication. The graph is then fed to three GNN models to detect malware in both supervised and unsupervised scenarios. Experiments demonstrate that these models significantly outperform the baseline model in various prediction tasks and require fewer data labels or less training data to do so.

Feng et al. [48] fed call graphs and functional dependency graphs into a GNN to accurately identify malware. They combined dynamic behavior and static structural features to improve sensitivity to code obfuscation and the inadequacy of traditional methods to represent features.

Yumlembam et al. [49] used a GNN and a VGAE-MalGAN adversarial architecture to detect malware on the Android platform. VGAE-MalGAN generated adversarial samples, and API graph embedding combined with permission and intention features improved the model's robustness and provided a more comprehensive feature representation. However, this approach still faces challenges related to dataset dependency and computational complexity in practical applications.

5.5 Resource Management

Network resource management involves efficiently allocating and using limited network resources. The goal is to ensure high performance, fairness, reliability, and scalability for applications and services. However, traditional approaches to resource management struggle with dynamic workloads, complex dependencies, high dimensionality, real-time requirements, and multi-objective optimization in today's networks. Since resource management is essentially a graph problem, GNNs are ideal for solving real-time, multi-objective resource management problems.

Chen et al. [50] proposed a GNN framework for network resource optimization in wireless IoT systems. The framework performs well in homogeneous systems and could potentially be used in heterogeneous networks.

Peng et al. [51] applied Vertex- and Edge-GNNs to learn network resource allocation policies. Both GNNs update their hidden representations by processing and pooling neighbor information to exploit topological information. The performance of the Vertex- and Edge-GNNs depends on the linearity and output dimensions of the processing and combination functions.

Li et al. [52] proposed TapFinger, a distributed scheduler that minimizes the total execution time of edge cluster tasks by co-optimizing task placement and fine-grained multi-resource allocation. TapFinger uses Multi-Agent Reinforcement Learning (MARL) to learn the uncertain resource sensitivity of the tasks and employs several techniques to improve efficiency.

Wang et al. [53] proposed an edge-update mechanism that enables GNNs to handle both node and edge variables, and proved its permutation equivariance property with respect to both transmitters and receivers.

Meng et al. [54] proposed a GNN-based algorithm that minimizes the total delay and energy consumption during training. This improves the performance of distributed FL in D2D wireless networks. The proposed GNN uses a multi-head graph attention mechanism to capture the various characteristics of clients and wireless channels. It also has a neighbor selection module that allows each client to select a subset of its neighbors to participate in model aggregation. A decoder is used for each client to determine the transmission power and computational resources.

Luo et al. [10] proposed a GNN-based resource allocation method to enhance a digital Twin's multiple Unmanned Aerial Vehicle (UAV) radar network. Through joint spectrum allocation and power control, this method maximizes the minimum signal-to-noise ratio (SINR) of all UAVs.

5.6 Extending to IoTs and IIoTs

Computer networks connect computers, routers, and servers for general-purpose communications, such as email, web browsing, and apps. IoTs connect sensors, actuators, and embedded devices for sensing and control. IIoTs connect Program Logic Controllers, Supervisory Control and Data Acquisition systems, robots, and industrial sensors for industrial system automation, monitoring, and control. IoT and IIoT systems differ from traditional computer networks in that they are heterogeneous, dynamic, and tightly coupled to the physical world. GNNs provide a powerful framework for learning from their graph-structured, spatio-temporal, and multi-modal data. This enables intelligent applications like fault detection, traffic forecasting, and secure control, especially in real-time industrial settings or at the edge.

Table 3. Scenarios and applications of GNNs in computer networks

<table>
<tr><th>Scenarios</th><th colspan="2">Applications</th><th>Contributions</th></tr>
<tr><td rowspan="2">Topo. Mana.</td><td colspan="2">TDA [34]</td><td>Improve representation of topologies with rings and holes.</td></tr>
<tr><td colspan="2">KerGNNs [35]</td><td>Combine graph kernel and GNN</td></tr>
<tr><td rowspan="2">Routing Opti.</td><td colspan="2">GNN+DRL [36-38]</td><td>Combine GNN and DRL</td></tr>
<tr><td colspan="2">[39]</td><td>Integrate graph structure and RL</td></tr>
<tr><td rowspan="4">Traffic Forecast</td><td colspan="2">Multi-time Scale Model [40]</td><td>Capture traffic abrupt changes at multiple scales</td></tr>
<tr><td colspan="2">T-ISTGNN [41]</td><td>RL and hypothesis transfer learning for domain adaptation</td></tr>
<tr><td colspan="2">TSGAN [42]</td><td>Time-series similarity-based GAT for cellular traffic prediction</td></tr>
<tr><td colspan="2">E2E Delay Modeling [43]</td><td>Model global network behavior correlations for load balancing</td></tr>
<tr><td rowspan="7">Cyber-security</td><td rowspan="4">Intru-sion</td><td>[7]</td><td>Edge Classification IoT Intrusion Detection System</td></tr>
<tr><td>E-ResGAT [44]</td><td>Residual connection improves class imbalance</td></tr>
<tr><td>DeepTraLog [45]</td><td>Combine logs and GNNs to detect RPC anomalies for microservices</td></tr>
<tr><td>Advers Attack [6]</td><td>Use saliency map and random walk to create adversarial samples</td></tr>
<tr><td rowspan="3">Mal-ware</td><td>Flow Graph [47]</td><td>Flow graph aggregates traffic behavior for malware detection.</td></tr>
<tr><td>Call Graph + FDG [48]</td><td>Use dynamic behavior and static features to improve obfuscation resilience</td></tr>
<tr><td>[49]</td><td>Adversarial architecture for Android malware detection.</td></tr>
<tr><td rowspan="6">Resource Mana.</td><td colspan="2">Resource Optimization [50]</td><td>Optimize resource allocation for wireless IoTs</td></tr>
<tr><td colspan="2">Resource Allocation [51]</td><td>Edge graph network to optimize resource allocation strategy</td></tr>
<tr><td colspan="2">TapFinger [52]</td><td>Jointly optimizE task placement and multi-resource allocation</td></tr>
<tr><td colspan="2">Edge-Update GNN [53]</td><td>Handle node/edge variables and prove permutation equivariability</td></tr>
<tr><td colspan="2">GNN+FL in D2D [54]</td><td>Use multi-head attention to reduce training cost</td></tr>
<tr><td colspan="2">UAV Radar Networks [10]</td><td>Maximize minimal UAV SINR to improve sensing coverage in multi-UAV systems.</td></tr>
<tr><td rowspan="7">Extending to IoTs & IIoTs</td><td colspan="2">Device Management [55]</td><td>Use a spatio-temporal graph to improve the collaboration of heterogeneous devices.</td></tr>
<tr><td colspan="2">Graph Embedding Anomaly Detection [56]</td><td>Real-time incremental learning and spatio-temporal correlation analysis to improve detection speed and accuracy</td></tr>
<tr><td colspan="2">[57]</td><td>IoT node classification.</td></tr>
<tr><td colspan="2">GAT-Based IoT ID [58]</td><td>Traffic graphs for accurate binary/multi-class classification</td></tr>
<tr><td colspan="2">EGNN [59]</td><td>Subgraphs mines device correlation, dual modes reduce energy cost</td></tr>
<tr><td colspan="2">Time-Series GGCN for Botnet Detection [60]</td><td>A time series polygon graph for traffic dynamic feature analysis</td></tr>
<tr><td colspan="2">HetEP [61]</td><td>Heterogeneous GNN fuses spatio-temporal relationship to predict manufacturing energy consumption.</td></tr>
</table>

Dong et al. [55] used GNNs to model the complex interactions between devices, thereby optimizing IoT device utilization and diagnosing system failures. They constructed a graph of the physical connection, communication topology, or data dependency between devices. Then, they abstracted a spatio-temporal graph and built a GAT model to improve the cooperation efficiency of heterogeneous devices for device management and fault diagnosis.

Jiang et al. [56] proposed a fast anomaly detection framework for IoT services. They used graph embedding to model the complex dependencies between logs. They designed a real-time incremental learning mechanism to handle dynamic data streams. They also employed spatio-temporal correlation analysis to effectively detect anomalies.

Sejan et al. [57] transformed IoT devices into graphs and used GNNs for node classification. They abstracted IoT devices as fully or randomly connected graphs and presented two GNN models (ARMAConv and Cluster-GCN) for experimentation.

Ahanger et al. [58] converted original network traffic into graphs and developed an IoT intrusion detection model based on GATs. Their experiments demonstrated 98% accuracy in binary classification and 99.2% accuracy in multi-classification.

Guo et al. [59] proposed an energy-efficient GNN (EGNN) for IoT anomaly detection. The EGNN introduced a subgraph generation approach for device association mining. It also adopted a dual-mode switching mechanism where only the central data of the subgraph was transmitted in normal mode and the entire subgraph was analyzed in abnormal mode to reduce energy consumption.

Altaf et al. [60] proposed an Internet of Things (IoT) botnet detection model based on a time-series Gated Graph Convolutional Network (GGCN). They improved detection accuracy on the Mirai dataset by up to 25% by constructing a time-series multilateral graph to analyze the dynamic characteristics of network traffic.

Su et al. [61] proposed a heterogeneous manufacturing correlation graph (HetMG) and an energy consumption prediction method (HetEP) based on HetMG. HetEP combines relational GCNs and LSTMs to identify spatio-temporal relationships among heterogeneous elements in the manufacturing process. It can predict order- and product-level energy consumption for green manufacturing based on heterogeneous GNNs.

All of the applications mentioned in this section are listed in Table 3.

6 Challenges and Open Issues

6.1 Graph Construction in Dynamic Networks

Graph construction transforms raw network data, such as system logs [62], packet traces, or topology snapshots [63], into a graph representation. Since the topology, traffic, and behavior evolve over time, graph construction remains a non-trivial in current research. Typical graph construction methods include snapshot-based, event-driven, sliding window temporal graphs, and learned or inferred graphs. However, updating graphs continuously from high-speed network streams while dealing with incomplete, noisy, or irregular traffic data remains a major challenge. Current research also lacks approaches for IoT and distributed systems, which may not report the full topology and face trade-offs between constructing fine-grained graphs and maintaining scalability. Future research directions include inferring dynamic graphs from unlabeled sequences and constructing fine-grained graphs only when anomalies or bottlenecks are detected, and federated/delayed graph construction and E2E frameworks, in which the GNN learns both the graph and the prediction task.

6.2 Spatio-Temporal Learning

Modeling spatio-temporal dynamics IN computer networks remains an open issue [64]. Modern architectures combine graph convolution with temporal modeling, often using RNNs, attention mechanisms, or diffusion mechanisms. However, these architectures struggle to represent graphs dynamically, handle asynchronous and bursty data, address label scarcity and delay feedback scenarios, and manage model complexity and computational overhead. Open issues include performing continuous learning on dynamic graphs with streaming data, designing real-time models with efficient temporal encoding and sparse GNN updates, handling multiscale patterns, and improving explainability.

6.3 Scalability to Large-Scale Networks

Applying GNNs to large-scale computer networks poses significant scalability challenges due to the explosive growth in graph size, dynamic updates, and real-time inference requirements [65]. Although sampling-based GNNs can limit computation and subgraph partitioning methods can divide large graphs into subgraphs for local training in mini-batch or distributed settings, controlling memory overhead and message passing costs for large networks remains challenging. Achieving real-time reference latency and topology updates for industrial applications is also a major issue. Future research directions include developing efficient online prediction models for edge systems, improving sample quality without full aggregation, and enabling visual, interpretable GNN inference for large, mission-critical networks.

6.4 Data Quality and Privacy in Sensitive Domains

Data quality and privacy are critical in sensitive domains such as healthcare, cybersecurity, and social networks. Trained GNN models can reveal sensitive node features or graph structures [66]. Incomplete or inaccurate graphs may result in missing edges, incorrect labels, or incomplete node features. Graphs distributed across institutions also raise data-sharing concerns. Currently, differential privacy is used to protect sensitive information. Federated GNNs, such as FLARE [67], use subgraphs to preserve utility while masking sensitive patterns [68]. Two potential future research directions are combining differential privacy and dynamic GNNs to provide a data-centric GNN pipeline and using generative models to produce privacy-preserving but utility-rich graphs and privacy-preserving collaborative modeling across distributed and heterogeneous IoT nodes.

6.5 Others

Future research directions include developing robust, trustworthy, and explainable GNNs in security-critical systems, training transferable GNNs that generalize across domains, network sizes, and topologies, and combining meta-learning or federated GNNs to handle data heterogeneity [69], and deploying GNNs in real-time systems with minimal disruption. Although Python scripts are often used for efficiency, tools such as Gephi [70] can be used for visualization. Publicly available traffic datasets include ISCX (www.unb.ca/cic/datasets/ vpn.html), USTC (staff.ustc.edu.cn/~cheneh/paper_pdf/ 2016/ang-Network-Traffic.pdf), CAIDA (catalog.caida.org), and CTU-13 (mcfp.weebly.com/the-ctu-13-dataset.html).

7 Conclusions

In this paper, we provide a systematic review of the basic concept of GNNs, discuss the advantages and disadvantages of their classical models, and summarize their typical use scenarios and applications in computer networks. We categorized six major usage scenarios, including topology management, routing optimization, traffic forecasting, cybersecurity, resource management, and extending to IoTs & IIoTs. Our focus was on the typical applications and how GNNs fit into them. We identified the challenges and open issues, and suggested the potential future research directions regarding emerging network scenarios and new technologies. By consolidating this rapidly growing yet fragmented research field, we have highlighted the unique value that GNNs bring to computer networks, serving as a foundational resource for both networking and machine learning research.

Acknowledgements

This work was supported by the National Natural Science Foundation of China (61962016), Natural Science Foundation of Guangxi (2023GXNSFAA026018), Guangxi Key Research and Development Program (AB22035018), GUAT Special Research Project on the Strategic Development of Distinctive Interdisciplinary Fields (TS2024441).

References

[1] F. Scarselli, M. Gori, A. C. Tsoi, M. Hagenbuchner, G. Monfardini, The Graph Neural Network Model, *IEEE Transactions on Neural Networks*, Vol. 20, No. 1, pp. 61-80, January, 2009.
https://doi.org/10.1109/TNN.2008.2005605

[2] Z. Wu, S. Pan, F. Chen, G. Long, C. Zhang, P. S. Yu, A Comprehensive Survey on Graph Neural Networks, *IEEE Transactions on Neural Networks and Learning Systems*, Vol. 32, No. 1, pp. 4-24, January, 2021.
https://doi.org/10.1109/TNNLS.2020.2978386

[3] X. Yang, K. Tang, H. Zhang, J. Cai, Auto-Encoding Scene Graphs for Image Captioning, *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Long Beach, California, USA, 2019, pp. 10677-10686.
https://doi.ieeecomputersociety.org/10.1109/CVPR.2019.01094

[4] A. Pareja, G. Domeniconi, J. Chen, T. Ma, T. Suzumura, H. Kanezashi, T. Kaler, T. Schardl, C. Leiserson, EvolveGCN: Evolving Graph Convolutional Networks for Dynamic Graphs, *Proceedings of the AAAI conference on artificial intelligence*, New York, USA, 2020, pp. 5363-5370.
https://doi.org/10.1609/aaai.v34i04.5984

[5] S. Wang, Y. Zheng, X. Jia, SecGNN: Privacy-Preserving Graph Neural Network Training and Inference as a Cloud Service, *IEEE Transactions on Services Computing*, Vol. 16, No. 4, pp. 2923-2938, July-August, 2023.
https://doi.org/10.1109/TSC.2023.3241615

[6] X. Zhou, W. Liang, W. Li, K. Yan, S. Shimizu, K. I.-K. Wang, Hierarchical Adversarial Attacks Against Graph-Neural-Network-Based IoT Network Intrusion Detection System, *IEEE Internet of Things Journal*, Vol. 9, No. 12, pp. 9310-9319, June, 2022.
https://doi.org/10.1109/JIOT.2021.3130434

[7] W. W. Lo, S. Layeghy, M. Sarhan, M. Gallagher, M. Portmann, E-GraphSAGE: A Graph Neural Network based Intrusion Detection System for IoT, *Proceeding of the 2022 IEEE/IFIP Network Operations and Management Symposium*, Budapest, Hungary, 2022, pp. 1-9.
https://doi.org/10.1109/NOMS54207.2022.9789878

[8] Y. Qiang, X. Wang, X. Liu, Y. Wang, W. Zhang, Edge-enhanced Graph Attention Network for driving decision-making of autonomous vehicles via Deep Reinforcement Learning, *Proceedings of the Institution of Mechanical Engineers, Part D: Journal of Automobile Engineering*, Vol. 239, No. 4, pp. 1168-1180, March, 2025.
https://doi.org/10.1177/09544070231217762

[9] D. Senapati, D. Maurya, A. Sarkar, C. Karfa, ERS: Energy-efficient Real-time DAG Scheduling on Uniform Multiprocessor Embedded Systems, *37th International Conference on VLSI Design and 23rd International Conference on Embedded Systems (VLSID)*, Kolkata, India, 2024, pp. 354-359.
https://doi.org/10.1109/VLSID60093.2024.00065

[10] J. Luo, Z. Fei, X. Wang, L. Zhao, B. Li, Y. Zhou, GNN-Based Resource Allocation for Digital Twin-Enhanced Multi-UAV Radar Networks, *IEEE Wireless Communications Letters*, Vol. 13, No. 11, pp. 3137-3141, November, 2024.
https://doi.org/10.1109/LWC.2024.3456247

[11] N. Shabani, J. Wu, A. Beheshti, Q. Z. Sheng, J. Foo, V. Haghighi, A. Hanif, M. Shahabikargar, A Comprehensive Survey on Graph Summarization with Graph Neural Networks, *IEEE Transactions on Artificial Intelligence*, Vol. 5, No. 8, pp. 3780-3800, August, 2024.
https://doi.ieeecomputersociety.org/10.1109/TAI.2024.3350545

[12] B. Khemani, S. Patil, K. Kotecha, S. Tanwar, A Review of Graph Neural Networks: Concepts, Architectures, Techniques, Challenges, Datasets, Applications, and Future Directions, *Journal of Big Data*, Vol. 11, No. 1, Article No. 18, January, 2024.
https://doi.org/10.1186/s40537-023-00876-4

[13] J. Vatter, R. Mayer, H.-A. Jacobsen, The Evolution of Distributed Systems for Graph Neural Networks and Their Origin in Graph Processing and Deep Learning: A Survey, *ACM Computing Surveys*, Vol. 56, No. 1, pp. 1-37, January 2024.

https://doi.org/10.1145/3597428
[14] M. Wu, M. Yan, W. Li, X. Ye, D. Fan, Y. Xie, Survey on Characterizing and Understanding GNNs From a Computer Architecture Perspective, *IEEE Transactions on Parallel and Distributed Systems*, Vol. 36, No. 3, pp. 537-552, March, 2025.
https://doi.org/10.1109/TPDS.2025.3532089
[15] B. M. Oloulade, J. Gao, J. Chen, T. Lyu, R. Al-Sabri, Graph neural architecture search: A survey, *Tsinghua Science and Technology*, Vol. 27, No. 4, pp. 692-708, August, 2022.
https://doi.org/10.26599/TST.2021.9010057
[16] R. Liu, P. Xing, Z. Deng, A. Li, C. Guan, H. Yu, Federated Graph Neural Networks: Overview, Techniques, and Challenges, *IEEE Transactions on Neural Networks and Learning Systems*, Vol. 36, No. 3, pp. 4279-4295, March, 2025.
https://doi.org/10.1109/TNNLS.2024.3360429
[17] S. Munikoti, D. Agarwal, L. Das, M. Halappanavar, B. Natarajan, Challenges and Opportunities in Deep Reinforcement Learning with Graph Neural Networks: A Comprehensive Review of Algorithms and Applications, *IEEE Transactions on Neural Networks and Learning Systems*, Vol. 35, No. 11, pp. 15051-15071, November, 2024.
https://doi.org/10.1109/TNNLS.2023.3283523
[18] M. Nandan, S. Mitra, D. De, GraphXAI: a survey of graph neural networks (GNNs) for explainable AI (XAI), *Neural Computing and Applications*, Vol. 37, No. 17, pp. 10949-11000, June, 2025.
https://doi.org/10.1007/s00521-025-11054-3
[19] F. Wang, Y. Liu, K. Liu, Y. Wang, S. Medya, P. S. Yu, Uncertainty in Graph Neural Networks: A Survey, *arXiv preprint*, arXiv: 2403.07185, March, 2024.
https://arxiv.org/abs/2403.07185
[20] X. Liu, J. Chen, Q. Wen, A Survey on Graph Classification and Link Prediction based on GNN, *arXiv preprint*, arXiv: 2307.00865, July, 2023.
https://arxiv.org/abs/2307.00865
[21] S. Wu, F. Sun, W. Zhang, X. Xie, B. Cui, Graph Neural Networks in Recommender Systems: A Survey, *ACM Computing Surveys*, Vol. 55, No. 5, pp. 1-37, May, 2023.
https://doi.org/10.1145/3535101
[22] G. Dong, M. Tang, Z. Wang, J. Gao, S. Guo, L. Cai, R. Gutierrez, B. Campbel, L. E. Barnes, M. Boukhechba, Graph Neural Networks in IoT: A Survey, *ACM Transactions on Sensor Networks*, Vol. 19, No. 2, pp. 1-50, May, 2023.
https://doi.org/10.1145/3565973
[23] S. Rahmani, A. Baghbani, N. Bouguila, Z. Patterson, Graph Neural Networks for Intelligent Transportation Systems: A Survey, *IEEE Transactions on Intelligent Transportation Systems*, Vol. 24, No. 8, pp. 8846-8885, August, 2023.
https://doi.org/10.1109/TITS.2023.3257759
[24] C. Chen, Y. Wu, Q. Dai, H.-Y. Zhou, M. Xu, S. Yang, X. Han, Y. Yu, A Survey on Graph Neural Networks and Graph Transformers in Computer Vision: A Task-Oriented Perspective, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 46, No. 12, pp. 10297-10318, December, 2024.
https://doi.org/10.1109/TPAMI.2024.3445463
[25] L. Wu, Y. Chen, K. Shen, X. Guo, H. Gao, S. Li, J. Pei, B. Long, Graph Neural Networks for Natural Language Processing: A Survey, *Foundations and Trends in Machine Learning*, Vol. 16, No. 2, pp. 119-328, January, 2023.
https://doi.org/10.1561/2200000096
[26] I. R. Ward, J. Joyner, C. Lickfold, Y. Guo, M. Bennamoun, A Practical Tutorial on Graph Neural Networks, *ACM Computing Surveys*, Vol. 54, No. 10s, pp. 1-35, January, 2022.
https://doi.org/10.1145/3503043
[27] L. Wu, P. Cui, J. Pei, L. Zhao, X. Guo, Graph Neural Networks: Foundation, Frontiers and Applications, *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, Washington, DC, USA, 2022, pp. 4840-4841.
https://doi.org/10.1145/3534678.3542609
[28] P. Reiser, M. Neubert, A. Eberhard, L. Torresi, C. Zhou, C. Shao, H. Metni, C. Hoesel, H. Schopmans, T. Sommer, P. Friederich, Graph Neural Networks for Materials Science and Chemistry, *Communications Materials*, Vol. 3, No. 1, Article No. 93, November, 2022.
https://doi.org/10.1038/s43246-022-00315-6
[29] Y. Hou, S. Jia, X. Lun, Z. Hao, Y. Shi, Y. Li, R. Zeng, J. Lv, GCNs-Net: A Graph Convolutional Neural Network Approach for Decoding Time-Resolved EEG Motor Imagery Signals, *IEEE Transactions on Neural Networks and Learning Systems*, Vol. 35, No. 6, pp. 7312-7323, June, 2024.
https://doi.org/10.1109/TNNLS.2022.3202569
[30] S. Xiao, S. Wang, Y. Dai, W. Guo, Graph Neural Networks in Node Classification: Survey and Evaluation, *Machine Vision and Applications*, Vol. 33, No. 1, Article No. 4, January, 2022.
https://doi.org/10.1007/s00138-021-01251-0
[31] T. R. Murgod, P. S. Reddy, S. Gaddam, S. M. Sundaram, C. Anitha, A Survey on Graph Neural Networks and its Applications in Various Domains, *Springer Nature Computer Science*, Vol. 6, No. 1, pp. 1-12, January, 2025.
https://doi.org/10.1007/s42979-024-03543-4
[32] T. N. Kipf, M. Welling, Variational Graph Auto-Encoders, *arXiv preprint*, arXiv: 1611.07308, November, 2016.
https://arxiv.org/abs/1611.07308
[33] J. Zhang, H. Zhang, C. Xia, L. Sun, Graph-Bert: Only Attention is Needed for Learning Graph Representations, *arXiv preprint*, arXiv: 2001.05140, January, 2020.
https://arxiv.org/abs/2001.05140
[34] M. Horn, D. E. Brouwer, M. Moor, Y. Moreau, B. Rieck, K. Borgwardt, Topological Graph Neural Networks, *arXiv preprint*, arXiv: 2102.07835, February, 2021.
https://arxiv.org/abs/2102.07835
[35] A. Feng, C. You, S. Wang, L. Tassiulas, KerGNNs: Interpretable Graph Neural Networks with Graph Kernels, *Proceedings of the AAAI Conference on Artificial Intelligence*, Virtual, 2022, pp. 6614-6622.
https://doi.org/10.1609/aaai.v36i6.20615
[36] A. Swaminathan, M. Chaba, D. K. Sharma, U. Ghosh, GraphNET: Graph Neural Networks for Routing Optimization in Software Defined Networks, *Computer Communications*, Vol. 178, pp. 169-182, October, 2021.
https://doi.org/10.1016/j.comcom.2021.07.025
[37] P. Almasan, J. Suárez-Varela, K. Rusek, P. Barlet-Ros, A. Cabellos-Aparicio, Deep Reinforcement Learning Meets Graph Neural Networks: Exploring a Routing Optimization Use Case, *Computer Communications*, Vol. 196, pp. 184-194, December, 2022.
https://doi.org/10.1016/j.comcom.2022.09.029
[38] B. Chen, D. Zhu, Y. Wang, P. Zhang, An Approach to Combine the Power of Deep Reinforcement Learning with A Graph Neural Network for Routing Optimization, *Electronics*, Vol. 11, No. 3, Article No. 368, February, 2022.

https://doi.org/10.3390/electronics11030368

[39] P. Tam, S. Ros, I. Song, S. Kang, S. Kim, A survey of Intelligent End-to-End Networking Solutions: Integrating Graph Neural Networks and Deep Reinforcement Learning Approaches, *Electronics*, Vol. 13, No. 5, Article No. 994, March, 2024.
https://doi.org/10.3390/electronics13050994

[40] S. Lin, H. Wang, Multi-Timescale Feature Fusion Enhanced Short-Term Network Traffic Burst Prediction, *2024 IEEE 12th International Conference on Information, Communication and Networks (ICICN)*, Guilin, China, 2024, pp. 240-245.
https://doi.org/10.1109/ICICN62625.2024.10761363

[41] Y. Qi, J. Wu, A. K. Bashir, X. Lin, W. Yang, M. D. Alshehri, Privacy-Preserving Cross-Area Traffic Forecasting in ITS: A Transferable Spatial-Temporal Graph Neural Network Approach, *IEEE Transactions on Intelligent Transportation Systems*, Vol. 24, No. 12, pp. 15499-15512, December, 2023.
https://doi.org/10.1109/TITS.2022.3215326

[42] Z. Wang, J. Hu, G. Min, Z. Zhao, Z. Chang, Z. Wang, Spatial-Temporal Cellular Traffic Prediction for 5G and Beyond: A Graph Neural Networks-Based Approach, *IEEE Transactions on Industrial Informatics*, Vol. 19, No. 4, pp. 5722-5731, April, 2023.
https://doi.org/10.1109/TII.2022.3182768

[43] W.-X. Liu, J. Cai, Y.-H. Zhu, J.-M. Luo, J. Li, Load Balancing inside Programmable Data Planes Based on Network Modeling Prediction Using A GNN with Network Behaviors, *Computer Networks*, Vol. 227, Article No. 109695, May, 2023.
https://doi.org/10.1016/j.comnet.2023.109695

[44] L. Chang, P. Branco, Graph-based Solutions with Residuals for Intrusion Detection: the Modified E-GraphSAGE and E-ResGAT Algorithms, *arXiv preprint*, arXiv: 2111.13597, pp. 11, November, 2021.
https://arxiv.org/abs/2111.13597

[45] H. X. Nguyen, S. Zhu, M. Liu, A Survey on Graph Neural Networks for Microservice-Based Cloud Applications, *Sensors*, Vol. 22, No. 23, Article No. 9492, December, 2022.
https://doi.org/10.3390/s22239492

[46] C. Yin, H. Zhang, M. Cheng, X. Xiao, X. Chen, X. Ren, P. Bogdan, Discovering Malicious Signatures in Software from Structural Interactions, *Proceeding of 2024 IEEE International Conference on Acoustics, Speech and Signal Processing*, Seoul, Korea, 2024, pp. 4845-4849.
https://doi.org/10.1109/ICASSP48485.2024.10446565

[47] J. Busch, A. Kocheturov, V. Tresp, T. Seidl, NF-GNN: Network Flow Graph Neural Networks for Malware Detection and Classification, *Proceeding of the 33rd International Conference on Scientific and Statistical Database Management*, Tampa, FL, USA, 2021, pp. 121-132.
https://doi.org/10.1145/3468791.3468814

[48] P. Feng, L. Gai, L. Yang, Q. Wang, T. Li, N. Xi, J. Ma, DawnGNN: Documentation Augmented Windows Malware Detection Using Graph Neural Network, *Computers & Security*, Vol. 140, Article No. 103788, May, 2024.
https://doi.org/10.1016/j.cose.2024.103788

[49] R. Yumlembam, B. Issac, S. M. Jacob, L. Yang, IoT-Based Android Malware Detection Using Graph Neural Network with Adversarial Defense, *IEEE Internet of Things Journal*, Vol. 10, No. 10, pp. 8432-8444, May, 2023.
https://doi.org/10.1109/JIOT.2022.3188583

[50] T. Chen, X. Zhang, M. You, G. Zheng, S. Lambotharan, A GNN-Based Supervised Learning Framework for Resource Allocation in Wireless IoT Networks, *IEEE Internet of Things Journal*, Vol. 9, No. 3, pp. 1712-1724, February, 2022.
https://doi.org/10.1109/JIOT.2021.3091551

[51] Y. Peng, J. Guo, C. Yang, Learning resource allocation policy: Vertex-GNN or edge-GNN?, *IEEE Transactions on Machine Learning in Communications and Networking*, Vol. 2, pp. 190-209, January, 2024.
https://doi.org/10.1109/TMLCN.2024.3354872

[52] Y. Li, X. Zhang, T. Zeng, J. Duan, C. Wu, D. Wu, X. Chen, Task Placement and Resource Allocation for Edge Machine Learning: A GNN-Based Multi-Agent Reinforcement Learning Paradigm, *IEEE Transactions on Parallel and Distributed Systems*, Vol. 34, No. 12, pp. 3073-3089, December, 2023.
https://doi.org/10.1109/TPDS.2023.3313779

[53] Y. Wang, Y. Li, Q. Shi, Y. -C. Wu, ENGNN: A General Edge-Update Empowered GNN Architecture for Radio Resource Management in Wireless Networks, *IEEE Transactions on Wireless Communications*, Vol. 23, No. 6, pp. 5330-5344, June, 2024.
https://doi.org/10.1109/TWC.2023.3325735

[54] C. Meng, M. Tang, M. Setayesh, V. W. S. Wong, Tackling Resource Allocation for Decentralized Federated Learning: A GNN-based Approach, *IEEE Transactions on Mobile Computing*, Vol. 24, No. 10, pp. 9554-9569, October, 2025.
https://doi.org/10.1109/TMC.2025.3562834

[55] G. Dong, M. Tang, Z. Wang, J. Gao, S. Guo, L. Cai, R. Gutierrez, B. Campbel, L. E. Barnes, M. Boukhechba, Graph Neural Networks in IoT: A Survey. *ACM Transactions on Sensor Networks*, Vol. 19, No. 2, pp. 1-50, May, 2023.
https://doi.org/10.1145/3565973

[56] X. Jiang, H. Luo, Y. Sun, M. Guizani, Fast Anomaly Detection for IoT Services Based on Multisource Log Fusion, *IEEE Internet of Things Journal*, Vol. 11, No. 6, pp. 9405-9419, March, 2024.
https://doi.org/10.1109/JIOT.2023.3323620

[57] M. A. S. Sejan, M. H. Rahman, M. A. Aziz, H.-K. Song, R. Tabassum, Adapting Graph Structure for IoT Network: A Performance Analysis, *Proceeding of the 21st International Joint Conference on Computer Science and Software Engineering*, Phuket, Thailand, 2024, pp. 74-78.
https://doi.org/10.1109/JCSSE61278.2024.10613682

[58] A. S. Ahanger, S. M. Khan, F. Masoodi, A. O. Salau, Advanced intrusion detection in internet of things using graph attention networks, *Scientific reports*, Vol. 15, No. 1, Article No. 9831, March, 2025.
https://doi.org/10.1038/s41598-025-94624-8

[59] H. Guo, Z. Zhou, D. Zhao, W. Gaaloul, EGNN: Energy-Efficient Anomaly Detection for IoT Multivariate Time Series Data Using Graph Neural Network, *Future Generation Computer Systems*, Vol. 151, pp. 45-56, February, 2024.
https://doi.org/10.1016/j.future.2023.09.028

[60] T. Altaf, X. Wang, W. Ni, G. Yu, R. P. Liu, R. Braun, GNN-Based Network Traffic Analysis for the Detection of Sequential Attacks in IoT, *Electronics*, Vol. 13, No. 12, Article No. 2274, June, 2024.
https://doi.org/10.3390/electronics13122274

[61] X. Su, M. Li, Q. Li, C. Chen, S. Guo, X. Wang, Energy Consumption Prediction for Manufacturing in Industrial IoT Based on Heterogeneous GNN, *Proceeding of the 2024*

IEEE/CIC International Conference on Communications in China, Hangzhou, China, 2024, pp. 627-632. https://doi.org/10.1109/ICCC62479.2024.10681781

[62] J. Li, H. He, S. Chen, D. Jin, J. Yang, LogGraph: Log Event Graph Learning Aided Robust Fine-Grained Anomaly Diagnosis, *IEEE Transactions on Dependable and Secure Computing*, Vol. 21, No. 4, pp. 1876-1889, July-August, 2024. https://doi.org/10.1109/TDSC.2023.3293111

[63] A. Caporali, K. Galassi, R. Zanella, G. Palli, GNN Topology Representation Learning for Deformable Multi-Linear Objects Dual-Arm Robotic Manipulation, *IEEE Transactions on Automation Science and Engineering*, Vol. 22, pp. 14738-14751, April, 2025. https://doi.org/10.1109/TASE.2025.3562231

[64] L. Van Langendonck, I. Castell-Uroz, P. Barlet-Ros, PPT-GNN: A Practical Pre-Trained Spatio-Temporal Graph Neural Network for Network Security, *arXiv preprint*, arXiv: 2406.13365, pp. 1-8, June, 2024. https://arxiv.org/abs/2406.13365

[65] S. Lyu, L. Peng, S. Y. Chang, Investigating Large-Scale RIS-Assisted Wireless Communications Using GNN, *IEEE Transactions on Consumer Electronics*, Vol. 70, No. 1, pp. 811-818, February, 2024. https://doi.org/10.1109/TCE.2023.3349153

[66] T. T. Mueller, M. Chevli, A. Daigavane, D. Rueckert, G. Kaissis, Privacy-utility trade-offs in neural networks for medical population graphs: Insights from differential privacy and graph structure, *arXiv preprint*, arXiv: 2307.06760, July, 2023. https://arxiv.org/abs/2307.06760

[67] X. Ma, J. Hu, S. Liang, Y. Wu, Federated Learning and Resource-Aware Graph Neural Network for Intrusion Detection in 6G-IoT Driven Healthcare System, *IEEE Internet of Things Journal*, pp. 1-1, April, 2025. https://doi.org/10.1109/JIOT.2025.3564722

[68] R. Zhang, Y. Chen, C. Wu, F. Wang, Cluster-driven GNN-based federated recommendation with biased message dropout, *2023 IEEE International Conference on Multimedia and Expo (ICME)*, Brisbane, Australia, 2023, pp. 594-599. https://doi.org/10.1109/ICME55011.2023.00108

[69] Y. Tian, L. Liu, J. Feng, Q. Pei, C. Chen, J. Du, C. Wu, Toward Robust and Generalizable Federated Graph Neural Networks for Decentralized Spatial-Temporal Data Modeling, *IEEE Transactions on Network and Service Management*, Vol. 21, No. 3, pp. 2637-2650, June, 2024. https://doi.org/10.1109/TNSM.2024.3386740

[70] M. Bastian, S. Heymann, M. Jacomy, Gephi: An Open Source Software for Exploring and Manipulating Networks, *3rd International AAAI Conference on Weblogs and Social Media*, San Jose, California, USA, 2009, pp. 361-362. https://doi.org/10.1609/icwsm.v3i1.13937

Biographies

Ling Xia Liao, full professor in the School of Artificial Intelligence, Guilin University of Aerospace Technology, China. Her research interests include intelligent network management and optimization, distributed systems, IoT, and edge computing.

Yulan Xu, graduate student in the School of Electronic Engineering and Automation, Guilin University of Electronic Technology, China. Her research interests include network security, management, and optimization.

Xuxia Liang, Chief physician and dean of the Obstetrics Department at the People's Hospital of Guangxi Zhuang Autonomous Region. Her research interests include high-risk pregnancy mornitoring and management, perinatal healthcare, and pregnancy complicated with reproductive tract malignancies.

Han-Chieh Chao, full professor in *Fo Guang* University, national Donghua University, Tamkang University, Taiwan, and UCSI University, *Malaysia*. His research interests include high speed networks, wireless networks, IPv6 based networks and digital divide, machine learning.

Shubao Pan, Senior Engineer in the School of Artificial Intelligence, Guilin University of Aerospace Technology. His research interests include machine learning, wireless networks, and photoelectric detection technology.