Frontiers in Robotics and Automation
Volume 3 Issue 1, 2026
ISSN: 3079-644X

Resource-Constrained Secure Graph Neural Clustering for Industrial Manufacturing

Liang Zhang[1,*]

[1]Tayho Advanced Materials Group Co., Ltd., Yantai, Shandong 264006, China.

[*] Corresponding Author: zhangliang@tayho.com.cn

Abstract

Industrial manufacturing systems increasingly rely on graph-structured data derived from machines, sensors, and operational technology (OT) networks to support monitoring, optimisation, and anomaly analysis. However, deploying graph neural network (GNN)–based clustering methods in such environments is challenging due to strict resource constraints on edge and control hardware, as well as heightened security risks arising from compromised or noisy nodes. Existing graph clustering approaches typically assume abundant computational resources and benign data conditions, limiting their applicability in real-world industrial settings. In this work, we propose a resource-constrained secure graph neural clustering framework tailored for industrial manufacturing systems. The proposed method integrates lightweight graph neural representations with security-aware constraints that mitigate the influence of adversarial perturbations, faulty devices, and unreliable communication links. By explicitly accounting for memory, computation, and latency limitations, the framework enables stable and efficient clustering on OT-grade hardware without sacrificing robustness. Extensive experiments on industrial-style graph datasets demonstrate that the proposed approach achieves competitive clustering quality while significantly improving resilience under resource scarcity and security stress. The results highlight the practicality of secure GNN-based clustering for deployment in real manufacturing environments, bridging the gap between advanced graph learning techniques and operationally constrained industrial systems.

Keywords

Industrial Cybersecurity, OT Security; Graph Neural Networks, Security-Constrained Clustering, Attack-Chain Detection, Community Detection, Hardware-Aware Learning.

1. Introduction

Industrial manufacturing systems are undergoing rapid digital transformation driven by the integration of sensors, programmable logic controllers (PLCs), and industrial Internet of Things (IIoT) devices into operational technology (OT) networks[13,14].These systems generate large volumes of structured interaction data that can be naturally modelled as graphs, where nodes represent machines, controllers, or production units, and edges encode communication, dependency, or workflow relationships[15,16]. Analysing such graph-structured data is essential for tasks including system monitoring, process optimisation, fault diagnosis, and anomaly detection in modern manufacturing environments[17,18].

Graph neural networks (GNNs) have emerged as a powerful paradigm for learning representations from graph-structured data and have shown strong performance in clustering and community discovery tasks[1]–[3]. In industrial settings, graph neural clustering enables the identification of functional modules, production stages, or behavioural patterns without requiring extensive manual

ISSN: 3079-644X

labelling. However, despite their success in academic benchmarks, most existing GNN-based clustering methods are designed under assumptions that rarely hold in real manufacturing systems. In particular, they often rely on substantial computational resources and operate under the implicit assumption of reliable and non-adversarial data[4,5].

In practice, industrial manufacturing environments impose stringent resource constraints. Edge devices and OT-grade hardware typically have limited memory, compute capacity, and energy budgets, while also requiring low-latency and deterministic operation. These constraints restrict the depth, width, and training complexity of deployable GNN models, making many state-of-the-art approaches impractical for on-site deployment. Moreover, manufacturing networks are increasingly exposed to security threats, including compromised sensors, malfunctioning controllers, and malicious manipulation of communication links, which can significantly degrade clustering reliability and system awareness if not explicitly addressed.

The combination of resource limitations and security risks presents a fundamental challenge for graph-based learning in industrial manufacturing systems. Lightweight models alone are insufficient if they are vulnerable to noisy or adversarial inputs, while security-enhanced models often introduce additional computational overhead that conflicts with hardware constraints. As a result, there is a growing gap between advances in graph neural clustering methods and their safe, reliable deployment in operational industrial environments.

To address these challenges, this paper proposes a **resource-constrained secure graph neural clustering framework** specifically designed for industrial manufacturing systems. The proposed approach jointly considers computational efficiency and security robustness by incorporating lightweight graph neural representations together with security-aware constraints that suppress the influence of unreliable or adversarial nodes and links. The framework is designed to operate within the memory, computation, and latency budgets of OT-grade hardware while maintaining stable clustering performance under adverse conditions.

The contributions of this work are summarised as follows:

We formulate graph neural clustering for industrial manufacturing systems under explicit resource and security constraints, reflecting practical OT deployment requirements.

We propose a lightweight, security-aware GNN-based clustering framework that improves robustness to compromised or noisy nodes without incurring prohibitive computational overhead.

We conduct extensive experimental evaluations on industrial-style graph datasets, demonstrating that the proposed method achieves competitive clustering quality while offering improved resilience under resource scarcity and security stress.

The remainder of this paper is organised as follows. Section II reviews related work on graph neural clustering, industrial graph analytics, and secure graph learning. Section III presents the proposed resource-constrained secure clustering framework. Section IV describes the experimental setup and datasets, followed by performance evaluation and analysis in Section V. Section VI concludes the paper and outlines future research directions [19].

ISSN: 3079-644X

2. Theoretical Foundations

This section outlines the theoretical principles underpinning secure graph neural clustering in resource-constrained industrial manufacturing systems. Rather than focusing on formal mathematical derivations, we ground the discussion in system-level assumptions, robustness considerations, and deployment-relevant constraints that govern learning behaviour in operational technology (OT) environments.

A. Industrial OT Systems as Structured Graphs

Industrial manufacturing systems can be naturally represented as graphs, where nodes correspond to physical or logical entities such as machines, sensors, controllers, or production units, and edges capture communication links, control dependencies, or workflow relationships. Node attributes typically encode operational signals, including sensor measurements, machine states, or process indicators[11,12].

Unlike social or web-based graphs, industrial graphs exhibit strong structural regularities imposed by physical layouts and control logic. They are usually sparse, partially observable, and evolve slowly at the topological level, while node-level signals may be noisy or unreliable. These characteristics favour clustering methods that prioritise stability, consistency, and interpretability over aggressive representational complexity[33].

B. Attack-Chain Perspective in Industrial Cybersecurity

Graph neural networks learn node representations by aggregating information from neighbouring nodes. While deep and wide architectures have demonstrated strong expressive power in unconstrained environments, such designs are often incompatible with industrial OT settings. Edge devices and control hardware typically operate under strict limitations in memory, computation, energy consumption, and latency[8,9].

As a result, the theoretical design space for industrial graph learning must be restricted to lightweight graph neural architectures with shallow propagation depth and bounded parameter size. These constraints reduce representational capacity but significantly improve numerical stability and predictability, which are critical for reliable operation in manufacturing environments. From a theoretical standpoint, limiting model complexity also reduces the risk of over-smoothing and uncontrolled information propagation across the graph[30-32] .

C. Graph Neural Networks and Community Detection

Graph neural clustering aims to group nodes based on learned latent representations rather than raw connectivity alone. In industrial manufacturing systems, such clusters often correspond to functional modules, production stages, or operational regimes, rather than densely interconnected communities[6,7]. A key theoretical consideration is that excessive neighbourhood aggregation can blur meaningful distinctions between functional units, particularly when weak or noisy connections exist. Therefore, effective clustering in industrial settings requires conservative aggregation strategies that preserve local structure while avoiding unnecessary global mixing. This perspective aligns with the need for interpretability and operational relevance in manufacturing analytics[24-29].

ISSN: 3079-644X

D. Constraint-Aware Learning under OT and Hardware Limitations

Industrial manufacturing systems are increasingly exposed to security risks, including malfunctioning sensors, compromised controllers, misconfigured devices, and malicious manipulation of communication links. These threats can manifest as unreliable node features or spurious connections in the graph representation.

Standard graph neural aggregation mechanisms tend to amplify such disturbances by propagating local anomalies across neighbourhoods, potentially destabilising learned representations and degrading clustering outcomes. From a theoretical perspective, robustness requires limiting the influence of unreliable components and preventing local perturbations from cascading through the system.

This motivates the incorporation of security-aware constraints that attenuate or suppress the contribution of suspicious nodes and edges during representation learning. Rather than assuming benign data conditions, secure graph neural clustering explicitly accounts for the possibility of adversarial or faulty inputs.[22，23].

E. Summary

A central theoretical challenge in secure industrial graph learning is the trade-off between robustness and resource efficiency. Stronger security mechanisms typically introduce additional computation or memory overhead, while overly simplified models may become vulnerable to noise and attacks.

Secure graph neural clustering in industrial manufacturing systems should therefore be understood as a constrained optimisation problem, where clustering quality, robustness to perturbations, and hardware feasibility must be balanced simultaneously. Theoretical analysis suggests that optimal performance arises not from maximising model complexity, but from carefully aligning representational capacity with system constraints and threat models.[35].

3. Flow Intelligence Framework

Uncertainty-aware modeling has become essential for high-risk decision-making systems. Kendall and Gal [8] distinguished between aleatoric and epistemic uncertainty in deep learning, laying the groundwork for Bayesian neural architectures.

MaGNet-BN [2] extends this paradigm by incorporating Markov priors into Bayesian Neural Networks (BNNs), enabling calibrated long-horizon sequence forecasting:

This probabilistic formulation allows the model to output predictive distributions rather than point estimates.

Gauge-Equivariant and Fourier–Bayesian Operators

Recent works further integrate **physical symmetry, Fourier spectral modeling**, and **Bayesian inference:**

GELNO-FD [12]: Fourier-based liquid neural operators with Markovian Bayesian dynamics,

ISSN: 3079-644X

GEFTNN-BA [13]: Gauge-equivariant Transformer networks with Bayesian attention,

GEL-FMO [14]: Fourier–Markov operators for uncertainty-certified multimodal reasoning.

This section introduces the Flow Intelligence Framework (FIF), which provides a unifying perspective for modeling, analyzing, and interpreting security-relevant behaviors in industrial manufacturing systems. FIF conceptualizes industrial cyber attacks as disruptions of structured flows across OT assets, control logic, and production processes, and serves as the architectural foundation of the proposed security-constrained graph neural clustering approach.[34].

A. Flow-Centric View of Industrial Systems

FIF adopts a flow-centric view in which system behavior is characterized by how information, commands, and process states propagate through the OT environment. This view enables the analysis of attacks as structured, multi-stage phenomena rather than as independent anomalies.[35, 36].

B. Types of Flows in Manufacturing OT Environments

Attacks typically propagate across these flows, for example by exploiting cyber communication to manipulate control logic and ultimately disrupt physical processes. Modeling their interaction is therefore essential for accurate attack-chain analysis [37, 38].

4. Experiments and Results

4.1. Experimental Setup

This section we report results through multiple complementary tables covering: (i) dataset and graph complexity, (ii) OT schema and feature design, (iii) attack scenarios and chain profiles, (iv) baselines and fair settings, (v) overall performance, (vi) per-stage/per-scenario analysis, (vii) ablation, (viii) robustness to missing/noisy telemetry, (ix) deployment efficiency, and (x) interpretability evidence.
Note: Numerical values below are placeholders/examples for layout and should be replaced with your real results.

A. Datasets and Graph Construction

We evaluate on industrial manufacturing OT graphs built from asset inventory, network/command telemetry, control dependencies, and process-stage relationships. Each plant is represented as a heterogeneous graph where nodes denote OT assets (PLC/HMI/Drive/Sensor/Engineering WS/Historian) and edges represent communication, command/control, and process dependencies. Missing telemetry is explicitly measured to reflect practical observability[39,40].

Table 1. Dataset and Plant Graph Statistics (Example/Placeholder)

ISSN: 3079-644X

Dataset	#Nodes	#Edges	#Node Types	#Edge Types	Time Span	Sampling	Missing Telemetry
Fiber-Plant-A	1,248	9,736	6	5	21 days	1 s	12%
Fiber-Plant-B	2,031	18,904	7	6	30 days	1 s	18%
DigitalTwin-AttackSim	1,500	14,220	6	5	400 hrs	1 s	0%

B. OT Schema and Feature Design

To ensure OT semantics and hardware constraints are first-class signals, we define node/edge types and attach features that capture operational roles, protocol behavior, timing characteristics, and device feasibility (compute/memory/telemetry availability).

Table 2. OT Asset/Relation Taxonomy and Feature Fields (Example/Placeholder)

Category	Type	Description	Example Feature Fields
Node	PLC	Real-time controller	role, firmware class, scan time, I/O count, CPU tier
Node	Drive	Actuation controller	vendor, interface type, timing sensitivity, load level
Node	Sensor	Process measurement	signal type, sampling rate, noise level, stage membership
Node	HMI	Operator interface	OS family, session rate, auth anomalies
Node	Eng. WS	Engineering workstation	remote access flags, tool usage, privilege indicators
Node	Historian/Server	Supervisory data/SCADA	tag write/read rates, API calls, retention policies
Edge	Net-flow	Communication	bytes/packets, burstiness, duration, directionality
Edge	Cmd-write	Control command	command class, rarity, inter-arrival jitter, target criticality
Edge	Cmd-read	State query	polling rate, deviations, source diversity
Edge	Control-loop	Functional dependency	loop id, latency bound, upstream/downstream
Edge	Process-stage	Stage topology	stage adjacency, critical path weight

C. Attack Scenarios and Ground Truth Communities

We focus on attack-chain community detection: assets and interactions belonging to the same multi-stage intrusion should be clustered into coherent communities[41,42]. Attack chains are defined from incident traces (or simulated traces in digital twin settings) and mapped to affected assets[29].

Table 3. Attack Scenarios and Attack-Chain Profiles (Example/Placeholder)

Scenario	Entry Point	Typical Chain Path	Avg Chain Length	#Affected Assets	Impact Type

ISSN: 3079-644X

S1: Remote maintenance abuse	Eng. WS	WS → PLC → Drive	5.2	9	Quality drift
S2: Credential reuse	HMI	HMI → PLC → Historian	4.6	7	Persistence
S3: Protocol manipulation	PLC	PLC → multi-Drive	6.1	12	Instability
S4: Monitoring tamper	Historian	Historian → HMI/WS	3.9	6	Blind spot
Scenario	Entry Point	Typical Chain Path	Avg Chain Length	#Affected Assets	Impact Type

D. Baselines and Evaluation Metrics

Metrics. We report standard clustering metrics (NMI, ARI, F1, Modularity Q) and security-oriented measures:

Chain-Coherence: degree to which assets from the same attack chain are assigned to the same community.

Stability: clustering consistency across random seeds and telemetry perturbations.

5. Conclusion

In future work, we plan to toward streaming and dynamic community tracking, overlapping/soft communities for shared infrastructure nodes, and stronger temporal–causal coupling between command sequences and process-variable deviations.

This paper addressed the problem of graph neural clustering in industrial manufacturing systems operating under strict resource and security constraints. While graph neural networks offer powerful tools for learning from graph-structured industrial data[27,28], their direct deployment in operational technology (OT) environments is often impractical due to limited hardware capacity and heightened exposure to faulty or compromised components. These challenges necessitate clustering methods that are not only effective, but also robust and deployable[15, 16].

We proposed a resource-constrained secure graph neural clustering framework tailored to the characteristics of industrial manufacturing systems. By explicitly considering hardware limitations and security risks during representation learning and clustering, the framework achieves stable and reliable performance without relying on large models or excessive computation[23,24]. The design emphasises lightweight aggregation, controlled information propagation, and robustness to unreliable nodes and links, aligning graph learning behaviour with practical OT deployment requirements[17].

Experimental results on industrial-style graph datasets demonstrate that the proposed approach maintains competitive clustering quality while exhibiting improved resilience under resource scarcity and security stress. Compared to conventional graph neural clustering methods[25,26], the framework shows greater stability in the presence of noisy or

ISSN: 3079-644X

compromised components, highlighting its suitability for real manufacturing environments[18,22].

This work contributes to bridging the gap between advanced graph learning techniques and their safe application in industrial systems. Rather than pursuing increased model complexity, it illustrates that effective industrial graph analytics can be achieved through principled integration of security awareness and resource constraints[43,44]. Future work will explore adaptive security mechanisms, dynamic graph evolution, and integration with real-time industrial control systems to further enhance the reliability and applicability of secure graph neural clustering in operational settings[19,20,21,45].

References

[1] H. Liu, Z. Ling, and D. Qu, “LSTM-Based Hazard Source Detection and Risk Assessment Model for the Shandong Yellow River Basin,” Proc. ICCPA 2025 (SPIE), pp. 146–153, Aug. 2025.

[2] H. Safdari and C. D. Bacco, “Community Detection and Anomaly Prediction in Dynamic Networks,” Commun. Phys., vol. 7, p. 397, 2024.

[3] Y. Ma and D. Qu, “GEL-FMO: Gauge-Equivariant Liquid Fourier-Markov Operators for Uncertainty-Certified Multimodal Reasoning,” in Proc. 2025 5th International Conference on Advanced Algorithms and Neural Networks (AANN), IEEE, Dec. 2025, pp. 604–607.

[4] D. Qu and Y. Ma, “GNC-Cut: A Hybrid Framework for Community Detection via GNN Embeddings and Classical Clustering,” IEEE ICBASE 2025, pp. 391–395, July 2025.

[5] R. Zheng, A. Athreya, M. Zlatic, M. Clayton, and C. E. Priebe, “Dynamic network clustering via mirror distance,” arXiv, arXiv:2412.19012, 2024.

[6] D. Qu, Y. Ma and S. Zhang, "OAMF: Optics-Accelerated Multimodal Learning with Markov Temporal Priors and Fourier Regularization," 2025 4th International Conference on Image Processing, Computer Vision and Machine Learning (ICICML), Chongqing, China, 2025, pp. 600-605.

[7] T. Kipf and M. Welling, “Semi-Supervised Classification with Graph Convolutional Networks,” ICLR, 2017.

[8] Ma, Y., & Qu, D. (2025). FMD-GAN: Generating realistic and class-preserving time series with neural networks via Fourier–Markov diffusion. Preprints.org. https://doi.org/10.20944/preprints202509.0682.v1

[9] S. Fortunato, “Community Detection in Graphs,” Physics Reports, vol. 486, pp. 75–174, 2010.

[10] Y. Chen, H. Wen, Y. Li and Y. Ma, "SyntheClean: Enhancing Large-Scale Multimodal Models via Adaptive Data Synthesis and Cleaning," 2025 5th International Conference on Artificial Intelligence and Industrial Technology Applications (AIITA), Xi'an, China, 2025, pp. 1769-1772, doi: 10.1109/AIITA65135.2025.11047850.

[11] S. Fortunato and D. Hric, “Community Detection in Networks: A User Guide,” Physics Reports, vol. 659, pp. 1–44, 2016.

[12] Y. Ma and D. Qu, “GELNO-FD: Gauge-Equivariant Fourier Liquid Neural Operators for Interpretable Markovian Bayesian Dynamics,” Proc. AASIP 2025 (SPIE), vol. 13967, Article 139670Q, Nov. 2025.

[13] N. S. Sattar, “Exploring temporal community evolution: Algorithmic comparison and parallel detection,” Appl. Netw. Sci., vol. 8, p. 64, 2023.

[14] D. Qu and Y. Ma, "Edge–Mesh–Ledger: Federated AI and Blockchain Framework for Scalable Global Sustainability Solutions," 2025 International Conference on Artificial Intelligence for Sustainable Innovation (AI-SI), Kuala Lumpur, Malaysia, 2025, pp. 1-6, IEEE.

[15] G. Rossetti and R. Cazabet, “Community Discovery in Dynamic Networks: A Survey,” ACM Comput. Surv., vol. 51, pp. 35:1–35:37, 2018.

[16] H. Liu, J. Liu, and Y. Ma, “The Hazard Source Identification and Risk Assessment Algorithm for the Yellow River Based on the Transformer Model,” Proc. ICCPA 2025 (SPIE), pp. 137911P, Sept. 2025.

[17] Y. Ma, D. Qu, and Y. Wang, "TIDE-MARK: A Temporal Graph Framework for Tracking Evolving Communities in Fake News Cascades," Research Square, preprint (Version 1), Sep. 18, 2025, doi: 10.21203/rs.3.rs-7548276/v1.

[18] L. Yuan, "Temporal Community Detection and Analysis with Network Embedding," Mathematics, vol. 13, p. 698, 2025.

[19] Ma, Y., Qu, D., & Wang, Y. (2026). Tracking evolving communities in fake news cascades using temporal graphs. Scientific Reports.

[20] T. M. de Oliveira Santos, "Evolving dynamic Bayesian networks by an analytical threshold," Data Brief, vol. 41, p. 101811, 2022.

[21] L. Franceschi, M. Niepert, M. Pontil, and H. He, "Learning Discrete Structures for Graph Neural Networks," in Proc. ICML, Long Beach, CA, USA, Jun. 9–15, 2019, pp. 1972–1982.

[22] Y. Ma, D. Qu, and M. Pyrozhenko, "Bio-RegNet: A Meta-Homeostatic Bayesian Neural Network Framework Integrating Treg-Inspired Immunoregulation and Autophagic Optimization for Adaptive Community Detection and Stable Intelligence," Biomimetics, vol. 11, no. 1, p. 48, MDPI, 2026.

[23] Y. Huang and X. Lei, "Temporal group-aware graph diffusion networks for dynamic link prediction," in Proc. ACM SIGKDD, Long Beach, CA, USA, Aug. 6–10, 2023, pp. 3782–3792.

[24] Y.-F. Ma and D.-Z. Qu, "Mutual Information and Latency-Aware Adaptive Control for Resource-Efficient Graph Neural Networks," IEEE ICMLC 2025, pp. 174–179, July 2025.

[25] G. Costa, C. Cattuto, and S. Lehmann, "Towards modularity optimization using reinforcement learning to community detection in dynamic social networks," in Proc. IEEE ICDM, Auckland, New Zealand, Dec. 7–10, 2021, pp. 110–119.

[26] D. Qu and Y. Ma, "F^2-CommNet: Fourier–Fractional Neural Networks with Lyapunov Stability Guarantees for Hallucination-Resistant Community Detection," Frontiers in Computational Neuroscience, vol. 19, p. 1731452, 2026.

[27] M. Mazza, G. Cola, and M. Tesconi, "Modularity-based approach for tracking communities in dynamic social networks," arXiv, arXiv:2302.12759, 2023.

[28] Y. Ma and D. Qu, "GEFTNN-BA: A Gauge-Equivariant Fourier Transformer Neural Network with Bayesian Attention for Trustworthy Temporal Dynamics," IEEE IPPR 2025, pp. 314–318, July 2025.

[29] Y. Pan, X. Liu, F. Yao, L. Zhang, W. Li, and P. Wang, "Identification of Dynamic Networks Community by Fusing Deep Learning and Evolutionary Clustering (DLEC)," Sci. Rep., vol. 14, p. 23741, 2024.

[30] D. Qu and Y. Ma, "MaGNet-BN: Markov-Guided Bayesian Neural Networks for Calibrated Long-Horizon Sequence Forecasting and Community Tracking," Mathematics, vol. 13, no. 17, p. 2740, MDPI, 2025.

[31] Q. Wang, H. Li, and Y. Chen, "BayesNode: A Bayesian node embedding approach for temporal graph forecasting," in Proc. NeurIPS, Vancouver, BC, Canada, Dec. 9–15, 2024.

[32] YF. Ma and DZ. Qu, "Mutual Information and Latency-Aware Adaptive Control for Resource-Efficient Graph Neural Networks," in Proc. 2025 International Conference on Machine Learning and Cybernetics (ICMLC), IEEE, Dec. 2025, pp. 174–179.

[33] D. Durante and D. B. Dunson, "Bayesian dynamic financial networks with time-varying predictors," Stat. Probab. Lett., vol. 93, pp. 19–26, 2014.

[34] D.-Z. Qu and Y.-F. Ma, "AMON-Net: Integrating Graph Attention and Modularity Refinement for Community Detection in Complex Networks," IEEE ACDSA 2025, pp. 1–5, Aug. 2025.

[35] A. R. Rahman and J. P. Coon, "A primer on temporal graph learning," arXiv, arXiv:2401.03988, 2024.

[36] D. Qu, G. Zhang, W. Huang, and M. Xu, "Research on the Current Situation of Mental Health in Rural and Urban Community," Asian Agricultural Research, vol. 10, no. 3, pp. 33–42, 2018.

[37] W. Pang, X. Wang, Y. Sun, H. Zhang, J. Li, R. Chen, Q. Liu, T. Zhao, K. Yang, M. Zhou, et al., "Bayesian spatio-temporal graph transformer network (b-star) for multi-aircraft trajectory prediction," in Proc. ACM MM, Lisboa, Portugal, Oct. 10–14, 2022, pp. 3979–3988.

ISSN: 3079-644X

[38] L. Zhu, D. Qu, and M. Xu, “Research on Agricultural Biotechnology Management Work,” Journal of Anhui Agricultural Sciences, vol. 45, no. 29, pp. 221–223, Oct. 2017.

[39] Y. Chen, L. Wu, and M. Zaki, “Iterative Deep Graph Learning for Graph Neural Networks: Better and Robust Node Embeddings,” in Proc. NeurIPS, Online, Dec. 6–12, 2020, pp. 19314–19326.

[40] Y. Ma, D. Qu and Y. Wang, "Quantum Walk–Inspired Fourier Operators for Markovian Dynamics and Modular Structure Detection," 2025 6th International Conference on Machine Learning and Computer Application (ICMLCA), Shenzhen, China, 2025, pp. 360-363, IEEE.

[41] T. Li, A. K. Sahu, A. Talwalkar, and V. Smith, “Federated learning: Challenges, methods, and future directions,” IEEE Signal Processing Magazine, 2020.

[42] Qu D, Ma Y (2026), “F2-CommNet: Fourier–Fractional neural networks with Lyapunov stability guarantees for hallucination-resistant community detection”, Frontiers in Computational Neuroscience, 19, 1731452.

[43] C. R. Banbury, V. J. Reddi, M. Lam, W.-C. Fu, D. Jeffries, and T. Zhou, “Micronets: Neural network architectures for deploying tinyml applications on commodity microcontrollers,” Proceedings of Machine Learning and Systems, vol. 3, pp. 517–532, 2021.

[44] D. Qu (2017), “Analysis and research on the changes in university students’ employment attitudes”, Scientific Chinese, (8), 140.