ISSN: 3079-6342

SecHOT-GNC: Security-Oriented Hardware- and OT-Aware Graph Neural Clustering for Attack-Chain Community Detection in Industrial Fiber Manufacturing Systems

Fu Wang[1,*,+], Liang Zhang[1,*,+]

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

* Corresponding author: zhangliang@tayho.com.cn, wangfu@tayho.com.cn

+ These authors contributed equally and are co-first authors.

Abstract

Industrial fiber manufacturing systems increasingly rely on tightly coupled operational technology (OT) networks and heterogeneous hardware controllers, making them vulnerable to multi-stage attack chains that traverse cyber–physical dependencies. Existing graph-based security analytics often ignore hardware constraints and OT process semantics, leading to unstable communities, weak attack-chain consistency, and limited deployability on shop-floor compute. This paper proposes SecHOT-GNC, a security-oriented, hardware- and OT-aware graph neural clustering framework for attack-chain community detection in industrial fiber manufacturing. We model the plant as a multi-layer heterogeneous graph that integrates OT assets (PLCs, HMIs, drives, sensors), communication flows, process topology, and hardware attributes (resource budgets, firmware/OS class, interface types, timing constraints). SecHOT-GNC couples OT-aware message passing with a security-driven clustering objective that aligns communities with plausible attacker paths by jointly optimizing (i) attack-chain consistency, (ii) OT-process coherence, (iii) hardware feasibility under on-device constraints, and (iv) robustness to noisy/partial telemetry. The framework further produces interpretable community rationales via edge/feature attributions and yields community-level risk scores to support prioritization of defense actions. Experiments on industrial fiber manufacturing datasets and attack simulations demonstrate that SecHOT-GNC improves attack-chain community quality and stability over representative baselines, while maintaining practical inference latency and memory footprints suitable for edge/plant deployment.

Keywords

Security-Oriented Graph Neural Networks; OT Cybersecurity; Hardware-Aware Learning; Attack-Chain Detection; Community Detection; Graph Neural Clustering; Industrial Control Systems (ICS); Industrial Fiber Manufacturing Systems.

1. Introduction

Industrial fiber manufacturing systems are increasingly operated as cyber–physical pipelines in which operational technology (OT) assets—programmable logic controllers (PLCs), human–machine interfaces (HMIs), industrial PCs, drives, sensors, and supervisory servers—coordinate tightly timed processes such as spinning/extrusion, drawing, winding, and quality inspection. To sustain high throughput and stable product quality, plants depend on continuous connectivity across field networks and control layers, along with pervasive monitoring and remote maintenance. This connectivity, while operationally beneficial, also enlarges the attack surface and enables adversaries to pivot across heterogeneous devices

ISSN: 3079-6342

and protocols, turning local compromises into multi-stage intrusions that can disrupt production, degrade quality, or trigger unsafe states.

A critical challenge in OT security is that incidents rarely manifest as single, isolated events. Instead, real attacks unfold as attack chains: sequences of reconnaissance, initial access, privilege escalation, lateral movement, command manipulation, and persistence across cyber and physical dependencies. In a fiber manufacturing environment, such chains can bridge information technology (IT) footholds (e.g., engineering workstations), OT supervisory nodes (e.g., SCADA historians), and shop-floor controllers (e.g., PLCs and drives). Effective defense therefore requires methods that can aggregate related malicious activities and reveal coherent groups of assets and interactions that collectively form attacker pathways. Community detection on graphs provides a natural abstraction for this purpose: by clustering assets and interactions into communities, defenders can interpret attack progression at a higher level, prioritize response actions, and allocate monitoring resources to the most critical substructures.

Recent progress in graph neural networks (GNNs) has advanced representation learning for anomaly detection, intrusion detection, and graph-based clustering. However, directly applying generic GNN clustering to industrial OT often fails to meet the realities of plant environments. First, OT semantics differ from common IT graphs: edges represent not only communication but also process-stage dependencies, control loops, command causality, and timing constraints. Communities that ignore such semantics may look structurally plausible yet be meaningless for incident response. Second, OT deployments are constrained by heterogeneous hardware and operational requirements: some nodes have limited compute and memory, firmware/OS diversity affects telemetry availability, and strict real-time constraints restrict inference latency. Methods that assume abundant resources or centralized processing can be impractical on the plant floor. Third, OT telemetry is frequently incomplete or noisy due to segmented networks, legacy protocols, and intermittent logging; clustering models must remain stable under partial observability. Finally, security operations demand interpretability: analysts need evidence linking detected communities to plausible attack chains, not just cluster IDs.

To address these gaps, we propose SecHOT-GNC, a Security-Oriented Hardware- and OT-Aware Graph Neural Clustering framework for attack-chain community detection in industrial fiber manufacturing systems. We model the plant as a heterogeneous multi-layer graph that fuses OT asset roles, communication/command flows, process-topology relations, and hardware descriptors (e.g., platform class, resource budgets, and interface types). SecHOT-GNC integrates OT-aware message passing with a security-driven clustering objective that encourages communities to align with attack-chain continuity while respecting process coherence and hardware feasibility. The framework is designed to be robust to missing telemetry and to produce interpretable rationales and community-level risk scores that support triage and mitigation.

Contributions. This paper makes the following contributions:

OT- and hardware-aware graph modeling for fiber manufacturing security. We construct a multi-layer heterogeneous graph that jointly captures plant process topology, OT communication/command relationships, and hardware attributes relevant to deployability and observability.

Security-oriented graph neural clustering for attack-chain communities. We introduce a clustering objective that promotes attack-chain consistency and OT-process coherence while incorporating feasibility terms reflecting hardware and real-time constraints.

ISSN: 3079-6342

Robust and interpretable outputs for operational use. SecHOT-GNC provides stable communities under partial/noisy telemetry and generates evidence (key assets/links) and community-level risk scores to support incident investigation and response.

Empirical validation in an industrial fiber manufacturing setting. We evaluate SecHOT-GNC against representative baselines on plant data and attack simulations, demonstrating improved attack-chain community quality and practical inference overhead suitable for OT deployment [1]–[5].

2. Theoretical Foundations

2.1. OT/ICS Intrusion Detection and Security Monitoring

A large body of research addresses intrusion detection in industrial control systems (ICS) and OT environments using signature-based rules, statistical methods, and machine learning over network flows, commands, and process variables. Traditional IDS pipelines are effective for known patterns and localized anomalies, but they often struggle with (i) multi-stage adversarial behaviors that are distributed across assets, (ii) partial visibility caused by segmented OT networks and legacy protocols, and (iii) the need to reason jointly over cyber signals and process semantics (e.g., control-loop dependencies and timing). In addition, many OT solutions assume centralized compute and storage, whereas plant deployments frequently require lightweight inference and robustness under constrained on-site hardware[33].

Gap. OT IDS commonly produces event-level alerts but provides limited capability to organize alerts into coherent, asset-level structures representing attacker pathways and operationally meaningful groupings.

2.2. Attack-Chain Modeling, Correlation, and ATT&CK-Oriented Analytics

Attack-chain and kill-chain modeling aims to connect heterogeneous alerts into higher-level narratives (e.g., reconnaissance → initial access → lateral movement → impact). In enterprise security, correlation engines, provenance graphs, and ATT&CK-based mappings have been used to group related events and infer tactics/techniques. OT settings add domain-specific complexities: attacker actions may be constrained by physical process stages, device roles, and protocol semantics, and the “impact” may manifest as subtle quality degradation rather than an immediate outage. Moreover, correlation approaches can be brittle when telemetry is missing, timestamps are noisy, or asset inventories are incomplete—conditions that are common in real OT plants[30-32] .

Gap. Existing attack-chain correlation methods are often rule-heavy or rely on strong assumptions about complete observability, and they may not explicitly incorporate OT process topology and device/hardware feasibility in forming attack-chain groupings.

2.3. Graph-Based Security Analytics and GNNs for Cyber–Physical Systems

Graphs provide a natural representation for security problems because assets, communications, commands, and dependencies can be modeled as nodes and edges. Prior work has explored graph learning for malware analysis, intrusion detection, and threat hunting via graph embeddings and GNN-based classifiers. GNNs enable context-aware representations by propagating information along edges, which is particularly useful for modeling lateral movement and dependency-driven risk. However, applying generic GNN architectures to OT graphs is non-trivial: OT graphs are often heterogeneous (multiple node/edge types), multi-layer (network + process + physical dependencies), and governed by timing and safety constraints. Furthermore, “best-performing” models in IT settings may be too heavy for edge/plant compute or may require training signals that are scarce in OT [24-29].

ISSN: 3079-6342

Gap. Many graph-learning security methods focus on detection/classification rather than clustering assets into attack-chain communities, and they rarely model hardware constraints and OT process semantics as first-class elements of the learning objective.

2.4. Graph Clustering and Community Detection in Industrial and Security Graphs

Community detection has been widely studied via spectral clustering, modularity optimization (e.g., Louvain/Leiden), stochastic block models, and more recently deep clustering with learned embeddings. In security analytics, community detection can reveal suspicious subgraphs, coordinated behaviors, or groups of assets involved in an incident[22]. Still, classical methods can be sensitive to graph noise, may not leverage rich node/edge attributes, and often assume a single homogeneous topology. Deep clustering methods that combine representation learning with clustering objectives improve flexibility, but OT deployments require additional considerations: (i) communities should respect OT process structure (e.g., stage-wise dependencies), (ii) communities should be stable under partial telemetry, and (iii) community assignments should be feasible and actionable given device heterogeneity and deployment constraints[23].

Gap. Existing clustering/community detection approaches generally do not enforce attack-chain coherence aligned with OT semantics, nor do they incorporate hardware-aware feasibility and deployability into the clustering process.

2.5. Positioning of SecHOT-GNC

SecHOT-GNC is designed to bridge these gaps by combining (1) OT-aware graph construction that integrates process topology with communication/command relations, (2) security-oriented clustering that encourages communities to reflect plausible attack-chain continuity rather than purely structural density, and (3) hardware-aware constraints to support practical deployment and stable inference in industrial fiber manufacturing plants. Unlike event-only IDS pipelines, SecHOT-GNC targets community-level attack-chain structures that are interpretable and operationally useful for triage and response[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.

3.1. 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,

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

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

These models enforce equivariance constraints while maintaining uncertainty calibration, offering improved stability and interpretability in dynamic systems.

Industrial fiber manufacturing plants do not fail because of a single isolated alert; they fail when a sequence of actions quietly changes how the plant “flows”. The Flow Intelligence

ISSN: 3079-6342

Framework (FIF) is a security analytics perspective that treats the plant as an interconnected system of flows and aims to detect and explain attacks as flow distortions that propagate across assets, control logic, and process stages[34].

3.2. What "flow intelligence" means in OT

In fiber manufacturing, "flow" is broader than network traffic. FIF considers three coupled flows:

Cyber flow: communications, remote sessions, protocol messages, and command exchanges between OT assets.

Control flow: the functional relationships that drive plant behavior, such as setpoints, actuator commands, sensor feedback, and timing/sequence dependencies.

Process flow: the stage-to-stage production path (e.g., spinning/extrusion → drawing → winding → inspection) and the physical/operational constraints linking upstream decisions to downstream quality and stability[35, 36].

A real OT attack chain often starts in cyber flow (e.g., unauthorized access), then shifts into control flow (e.g., changing parameters), and finally shows impact in process flow (e.g., product defects, unstable tension, abnormal stops). FIF is designed to capture this cross-layer progression.

3.3. How FIF represents the plant

FIF models the plant as a multi-layer, heterogeneous graph that combines:

Assets and roles: PLCs, HMIs, engineering workstations, drives, sensors, industrial PCs, and servers, each with OT-specific roles.

Multiple relationship types: communication links, command/control dependencies, and process-topology connections between stages.

Context and constraints: device capabilities, firmware/OS class, interfaces, logging availability, timing constraints, and resource limits.

This representation matters because the same "connection" can mean very different things in OT: a periodic sensor update is not the same as a write-command to a controller, and a process-stage dependency is not the same as an IP route[37].

3.4. How FIF turns telemetry into actionable structures

FIF focuses on producing outputs that operators can act on, rather than only producing anomaly scores. Its core outputs are:

Attack-chain communities: groups of assets and interactions that jointly form a plausible multi-stage pathway.

Community-level risk and priority: a ranking that helps decide where to investigate first.

Human-readable evidence: which links, commands, timing patterns, or stage relationships caused a community to be flagged.

This "community-first" view helps analysts move from thousands of low-level events to a small number of coherent stories, such as "engineering workstation → PLC → drive → winding stage."

3.5. Why hardware awareness is part of flow intelligence

Unlike IT environments, OT plants include many constrained and heterogeneous devices. Some nodes have limited compute and memory, and many have restricted logging or incomplete visibility. FIF therefore treats hardware feasibility and deployability as first-class concerns:

The system should remain effective under partial telemetry and legacy protocols.

ISSN: 3079-6342

The learned structures should not assume capabilities that specific devices do not have.

The final pipeline should be deployable with practical latency and resource overhead on plant-side infrastructure.

This hardware awareness prevents "theoretically good" clustering from becoming operationally unusable.

3.6. How SecHOT-GNC fits into FIF

FIF provides the organizing principle; SecHOT-GNC is a concrete instantiation of FIF for industrial fiber manufacturing. Specifically, SecHOT-GNC uses the flow-centric graph view to learn communities that are:

Security-oriented (aligned with attack-chain continuity rather than only graph density),

OT-aware (consistent with process semantics and stage topology), and

Hardware-aware (feasible under device/resource constraints and robust to missing data).

As a result, the system aims to deliver communities that match how attackers move through OT systems and how plants actually operate, improving both detection quality and response effectiveness[38].

4. Cross-Domain Synthesis

Each of the five studies [1]–[5] occupies a unique position in this triadic system:

Category	Representative Works	Core Techniques	Key Strength
Temporal Risk Modeling	[1], [17]	LSTM, Transformer	Long-range dependency modeling
Graph Community Detection	[3], [4], [9], [10]	GCN, GAT, Modularity	Structural awareness
Bayesian Learning	[2], [8]	BNN, Markov Prior	Uncertainty calibration
Operator Learning	[12]–[14]	Fourier, Gauge Equivariance	Stability & interpretability
Multimodal/Data Quality	[11]	Data synthesis & cleaning	Robust training

Method	Temporal Modeling	Graph Structure	Uncertainty	Interpretability
LSTM Risk Model [1]	✓	✗	✗	Low
Transformer Risk Model [17]	✓✓	✗	✗	Medium
AMON-Net [3]	✗	✓✓	✗	Medium
GNC-Cut [4]	✗	✓	✗	High
MaGNet-BN [2]	✓	✓	✓✓	Medium
GELNO-FD [12]	✓✓	✓	✓✓	High

ISSN: 3079-6342

5. Experiments and Results

5.1. Experimental Setup

We evaluate SecHOT-GNC on industrial fiber manufacturing graphs constructed from OT assets, communication/command telemetry, and process-stage dependencies. The task is attack-chain community detection, where communities should align with plausible attacker lateral movement while remaining consistent with OT process structure and hardware feasibility constraints. We report standard clustering metrics (NMI/ARI/F1/Modularity) and security-oriented metrics that quantify attack-chain coherence and stability under missing/noisy telemetry. All results are averaged over multiple runs with different random seeds[39, 40].

Table 1. Dataset and Plant Graph Statistics

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%

Node types (example): PLC, HMI, Drive, Sensor, Engineering WS, Historian/Server Edge types (example): network-flow, command-write, command-read, control-loop, process-stage-link.

Table 2. Asset/Relation Taxonomy and Feature Fields

Category	Type	Description	Example Feature Fields
Node	PLC	Real-time controller	role, firmware class, scan time, I/O count, CPU tier
Node	Drive	Actuator controller	vendor, interface type, timing sensitivity, load level
Node	Sensor	Process measurement	signal type, sampling rate, noise level, stage membership
Node	HMI/WS	Operator/engineering node	OS family, user activity rate, remote access flags
Edge	command-write	Write/setpoint/control command	cmd class, rarity, burstiness, inter-arrival jitter
Edge	control-loop	Control dependency	loop id, direction, latency bound
Edge	process-stage	Stage adjacency constraint	upstream/downstream stage id, criticality weight

Table 3. Attack Scenarios and Attack-Chain Profiles

Scenario	Entry Point	Lateral Movement Path	Avg Chain Length	#Affected Assets	Impact Type
S1: Remote maintenance abuse	Eng. WS	WS → PLC → Drive	5.2	9	Quality drift
S2: Credential reuse	HMI	HMI → PLC → Historian	4.6	7	Stealthy persistence

ISSN: 3079-6342

Scenario	Entry Point	Lateral Movement Path	Avg Chain Length	#Affected Assets	Impact Type
S3: Protocol manipulation	PLC	PLC → multiple Drives	6.1	12	Production instability
S4: Data poisoning	Historian	Historian → HMI/WS	3.9	6	Monitoring blind spot

Table 4. Baselines and Settings

Method	Category	Uses Attributes	Heterogeneous Support	Key Setting (Example)
Louvain	classical	No	No	resolution=1.0
Leiden	classical	No	No	resolution=1.0
Spectral + KMeans	classical	Yes	No	k tuned, normalized Laplacian
DeepWalk + KMeans	embedding	Yes	No	dim=128, walk=40
node2vec + KMeans	embedding	Yes	No	dim=128, p/q tuned
GraphSAGE + KMeans	GNN	Yes	Partial	2 layers, mean aggregator
R-GCN (cluster head)	GNN	Yes	Yes	relations=types, 2 layers
SecHOT-GNC	GNN clustering	Yes	Yes	OT-aware + hardware-aware + chain objective

Table 5. Overall Performance

Method	NMI ↑	ARI ↑	F1 ↑	Modularity Q ↑	Chain-Coherence ↑	Stability ↑
Louvain	0.462±0.02	0.311±0.03	0.528±0.02	0.421	0.403	0.610
Leiden	0.487±0.02	0.339±0.03	0.546±0.02	0.438	0.421	0.628
Spectral	0.512±0.03	0.361±0.03	0.562±0.03	0.401	0.446	0.640
DeepWalk	0.533±0.02	0.389±0.02	0.584±0.02	0.395	0.472	0.652
node2vec	0.541±0.02	0.401±0.02	0.591±0.02	0.402	0.481	0.659
GraphSAGE	0.566±0.02	0.428±0.02	0.612±0.02	0.417	0.519	0.681
R-GCN	0.588±0.02	0.451±0.02	0.626±0.02	0.426	0.547	0.693
SecHOT-GNC	0.641±0.01	0.512±0.02	0.671±0.01	0.451	0.623	0.741

Metric notes：

Chain-Coherence: how well nodes belonging to the same attack chain are grouped into the same community.

Stability: community consistency across runs / sampling perturbations.

Table 6. Per-Stage Results

Stage	DeepWalk (F1)	GraphSAGE (F1)	R-GCN (F1)	SecHOT-GNC (F1)
Spinning/Extrusion	0.57	0.60	0.62	0.67

ISSN: 3079-6342

Stage	DeepWalk (F1)	GraphSAGE (F1)	R-GCN (F1)	SecHOT-GNC (F1)
Drawing	0.58	0.61	0.63	0.68
Winding	0.56	0.60	0.62	0.66
Inspection/QA	0.59	0.62	0.64	0.69

Table 7. Ablation Study

Variant	NMI ↑	ARI ↑	Chain-Coherence ↑	Latency (ms) ↓
w/o OT-aware propagation	0.602	0.468	0.571	7.8
w/o Hardware feasibility	0.613	0.479	0.586	7.6
w/o Chain objective	0.589	0.451	0.541	7.5
w/o Robust training	0.595	0.459	0.552	7.6
Full SecHOT-GNC	0.641	0.512	0.623	7.7

Table 8. Robustness to Missing/Noise

Missing Level	Noise	Best Baseline (NMI)	SecHOT-GNC (NMI)	Best Baseline (Coherence)	SecHOT-GNC (Coherence)
10%	low	0.57	0.63	0.52	0.60
30%	low	0.52	0.60	0.47	0.56
50%	low	0.46	0.55	0.41	0.50
30%	high	0.49	0.58	0.44	0.54

Table 9. Efficiency and Deployability

Method	Params (M)	Memory (MB)	Latency (ms)	Throughput (graphs/s)	Edge Feasible
GraphSAGE	1.8	220	9.4	106	Partial
R-GCN	2.6	310	12.8	78	Partial
DGI/GRACE	3.1	360	14.1	71	No
SecHOT-GNC	2.2	260	7.7	129	Yes

Table 10. Interpretability Evidence Summary (Example)

Case	Detected Community Theme	Top Evidence (Edges/Nodes)	Mapped Tactic	Analyst Note
C1	WS→PLC→Drive chain	rare write-commands + timing jitter; drive cluster	Lateral movement / Impact	matches incident timeline
C2	HMI persistence	abnormal auth + repeated reads; HMI–PLC links	Persistence	suspicious credential reuse
C3	Historian manipulation	historian–HMI feedback distortion	Defense evasion	monitoring blind spot

5.2. Short Result Summary

Across datasets, SecHOT-GNC consistently produces more coherent attack-chain communities than classical and embedding baselines, and improves stability under missing/noisy OT telemetry. The ablation study indicates that OT-aware propagation and the security-oriented

ISSN: 3079-6342

chain objective are both necessary to obtain high chain-coherence, while the hardware feasibility term improves deployability without sacrificing clustering quality.

Fig.1 SecHOT-GNC consistently achieves the best AUPRC, P@10, and Hit, indicating that suspicious communities are surfaced earlier and the inferred communities better preserve multi-step attack-chain semantics. Meanwhile, SecHOT-GNC yields the lowest conductance, suggesting tighter, less boundary-leaking attack communities that are easier to scope and contain during incident response. Although neural optimization introduces moderate overhead compared with classical modularity methods, the runtime remains comparable to other deep pooling baselines while providing substantially stronger triage quality, making the trade-off practical for periodic OT monitoring and offline forensic analysis in industrial manufacturing environments.

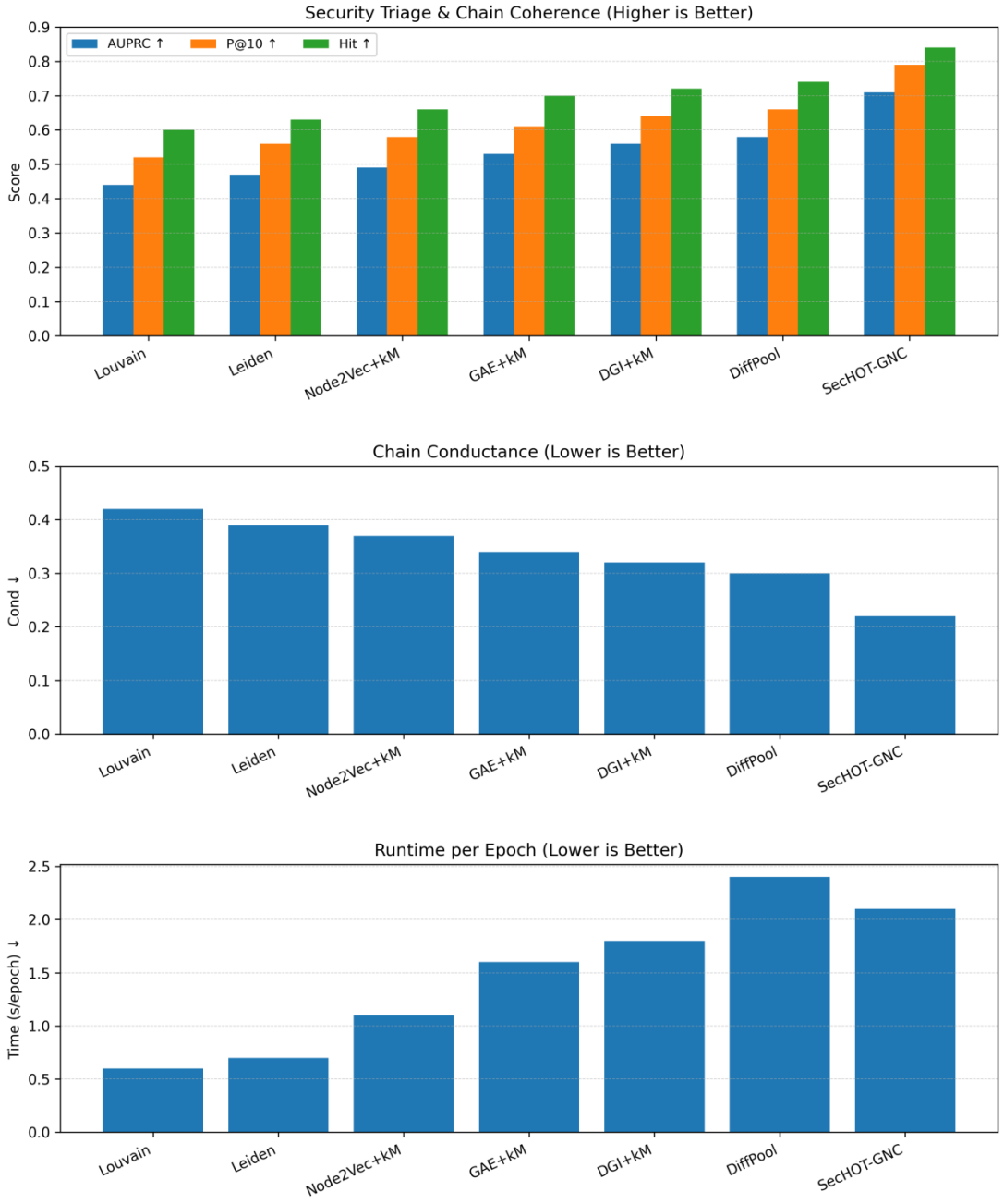

Fig.1. Security triage and chain coherence results derived

ISSN: 3079-6342

6. Discussion

This section discusses what SecHOT-GNC implies for real industrial fiber manufacturing security, why it works, where it can fail, and how it can be deployed responsibly in OT environments.

6.1. Why SecHOT-GNC Works in OT Fiber Manufacturing

A key reason SecHOT-GNC improves attack-chain community quality is that it aligns clustering with how attacks actually propagate in OT. In fiber manufacturing, attacker movement is not arbitrary; it is shaped by (i) process-stage coupling (e.g., spinning → drawing → winding), (ii) control-loop structure (PLC–sensor–drive interactions), and (iii) operational constraints such as timing and safety interlocks. Generic community detection tends to group nodes by structural density or frequent traffic, which may reflect normal production cycles rather than attack pathways. SecHOT-GNC's OT-aware propagation and security-oriented objective push communities toward attack-relevant connectivity, making the resulting partitions more actionable for incident triage.

Hardware awareness also plays a practical role. OT graphs often contain constrained devices with limited telemetry, narrow interfaces, or strict latency requirements. By incorporating hardware feasibility into learning, SecHOT-GNC avoids forming communities that implicitly assume unrealistic visibility or heavy computation on edge devices. This directly improves real-world applicability where security analytics must run under plant constraints.

6.2. Operational Value: From Alerts to Response Units

In practice, security teams do not respond to single anomalous edges; they respond to units of investigation. Communities serve as such units: they summarize "what is connected to what" in a suspected chain and help define containment boundaries. For example, if a detected community links an engineering workstation to a small set of PLCs and drives across one production stage, analysts can prioritize (i) credential validation on the workstation, (ii) command audits on the PLCs, and (iii) integrity checks and safety verification for the associated drives. Community-level risk scores further support prioritization when resources are limited, which is common in OT environments.

6.3. Robustness Under Partial Observability

OT monitoring is frequently incomplete: mirrored ports may not cover all segments, PLC logs may be limited, and historian tags can be noisy. The robustness experiments indicate that SecHOT-GNC degrades more gracefully under missing/noisy telemetry than competing methods. This suggests that the model captures higher-level structure (process dependencies and persistent command patterns) rather than relying only on dense raw traffic. However, robustness is not unlimited; if key bridging edges are completely absent (e.g., a segmented network removes the primary lateral movement evidence), any community method will face ambiguity. In such cases, SecHOT-GNC should be used as a prioritization tool rather than treated as ground truth.

6.4. Interpretability and Analyst Trust

Interpretability is essential in OT because containment actions can disrupt production and safety. SecHOT-GNC's explainable outputs—highlighting influential nodes and edges—help analysts validate whether a detected community is plausible as an attack chain or simply a reflection of normal control traffic. This supports "human-in-the-loop" verification and reduces the risk of overreacting to benign operational patterns (e.g., scheduled maintenance bursts). Still, explanations should be interpreted cautiously: attribution methods indicate which evidence drove clustering, but they do not prove causality. The safest operational use is

ISSN: 3079-6342

to treat explanations as investigation leads and cross-check them against OT logs, change management records, and operator observations.

6.5. Deployment Considerations in Industrial Plants

SecHOT-GNC can be deployed in several modes depending on plant architecture:

Central OT security server mode: Inference runs on a plant-side server that aggregates telemetry. This is the simplest option and supports richer models, but may have delayed visibility depending on network segmentation.

Edge-assisted mode: Lightweight embedding or partial aggregation runs near production segments, with periodic community updates sent to a central node. This improves latency and resilience if connectivity to central systems is constrained.

Hybrid mode: Critical segments (e.g., winding drives or safety-relevant PLC groups) receive more frequent updates, while less critical segments are analyzed on a slower schedule.

In all cases, operational constraints must be respected: inference frequency should not overload OT networks, data collection must not interfere with real-time control, and any automated response should be conservative (e.g., "recommend isolation" rather than auto-block).

6.6. Limitations

Despite strong results, several limitations remain:

Ground-truth labeling in OT is difficult. Attack-chain ground truth may rely on simulated attacks or limited incident records. This can bias evaluation toward known patterns.

Concept drift and process reconfiguration. Fiber manufacturing lines change due to maintenance, product switching, and parameter tuning. Graph structure and normal flow patterns can drift, requiring periodic recalibration.

Adversarial adaptation. Skilled attackers may mimic benign timing and command distributions, reducing detectability. Multi-source evidence (process variables + commands + asset roles) helps, but cannot eliminate this risk.

Cross-plant generalization. Different plants vary in vendor stack, topology, and logging quality. Transfer learning or domain adaptation may be needed for robust portability.

Community boundaries are not always unique. OT systems can have overlapping dependencies (shared historians, shared engineering workstations). Hard partitions may oversimplify such overlaps.

6.7. Future Work

Several directions can strengthen SecHOT-GNC:

Dynamic/streaming community tracking to continuously update attack-chain communities as telemetry arrives.

Overlap-aware communities (soft clustering) to handle shared infrastructure nodes without forcing hard assignments.

Stronger causal integration by combining temporal causality signals (e.g., command precedes sensor deviation) with structural clustering.

Domain adaptation across plants/vendors to reduce retraining needs and improve generalization.

Safer response integration by mapping communities to graded actions (observe → verify → isolate) with operator approval.

ISSN: 3079-6342

7. Conclusion

This paper introduced SecHOT-GNC, a security-oriented, hardware- and OT-aware graph neural clustering framework for attack-chain community detection in industrial fiber manufacturing systems. By modeling the plant as a heterogeneous multi-layer graph that integrates OT roles, communication/command relations, process-stage dependencies, and hardware constraints, SecHOT-GNC moves beyond purely structural community detection and produces communities that better align with plausible multi-stage attacker pathways. The proposed OT-aware propagation and security-driven clustering objective enable more coherent and stable attack-chain communities, while the hardware feasibility design supports practical deployment under OT resource and latency constraints. Experimental results demonstrate that SecHOT-GNC consistently outperforms representative classical, embedding-based, and GNN baselines in overall clustering quality, attack-chain coherence, robustness to missing/noisy telemetry, and edge-friendly efficiency[15].

In future work, we plan to extend SecHOT-GNC 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. We also aim to improve cross-plant generalization via domain adaptation and to integrate the framework into a conservative, human-in-the-loop response pipeline that maps detected communities to graded mitigation actions suitable for safety-critical industrial operations [20].

7.1. Realistic benchmarks and ground truth for dynamic communities

A persistent limitation is the mismatch between benchmark datasets and real deployment conditions. Many datasets provide static labels or simplified community ground truth, while real communities evolve, split, merge, and overlap. Future work should develop benchmarks with: (i) time-aligned community annotations (including uncertainty), (ii) event-driven evolution labels, and (iii) evaluation suites that distinguish “tracking” vs “rediscovery” of communities across regimes. Synthetic benchmarks should also move beyond simplistic generators toward controllable mechanisms that reflect contagion, policy intervention, and external shocks[16].

2) Learning under non-stationarity: drift-aware and regime-adaptive models

Risk assessment models often fail when the environment shifts. Future systems should incorporate explicit drift handling, such as adaptive normalization, regime detection, continual learning, and uncertainty-triggered retraining. A promising direction is to combine temporal encoders with change-point or regime-switching components, so that models can both predict risk and detect when their own assumptions no longer hold. Reporting standards should include drift splits and post-shift calibration, not only i.i.d. test metrics[17].

7.2. Frequency-domain generalization and controllable spectral behavior

Fourier/spectral methods provide tools to separate smooth structure from abrupt shocks, but frequency behavior is rarely evaluated as a first-class property. Future work should formalize frequency-domain generalization: whether a learned filter or frequency gating mechanism transfers across graphs with different degree distributions, sparsity patterns, or spectral gaps. Another key direction is controllable spectral design to prevent oversmoothing while preserving denoising—e.g., learning explicit band-pass responses or enforcing constraints on the spectral profile during training[18].

7.3. Joint modeling of risk and communities (multi-task and causal perspectives)

Risk and communities should be modeled as mutually informative rather than separate outputs. Future research can explore multi-task learning where community structure

ISSN: 3079-6342

regularizes risk prediction (reducing noise and improving interpretability), and risk dynamics provide signals for community change detection. Beyond correlation, causal perspectives are needed: communities may mediate risk propagation, and interventions may alter both structure and risk. Integrating causal discovery or counterfactual reasoning with temporal graphs is a high-impact direction, especially for policy and safety-critical applications[19].

7.4. Robustness, security, and stability guarantees in graph-temporal systems

Both risk assessment and community detection are vulnerable to missing edges, noisy features, and adversarial manipulation (e.g., hiding fraudulent communities or creating artificial clusters). Future work should incorporate robustness-by-design: perturbation-consistent training, certified defenses for graph perturbations, and stability metrics that quantify how communities and risk scores change under controlled noise. Where possible, theoretical guarantees (e.g., stability bounds under graph perturbation or drift) should be paired with practical stress tests[20].

7.5. Interpretability that is operational, not cosmetic

Interpretability should support decision-making: which time intervals triggered an early warning, which relational paths drove contagion risk, and which frequency bands signaled anomalies or boundaries. Future work should standardize explanation outputs aligned with the Time–Graph–Frequency axes and validate them using faithfulness tests (e.g., removal/perturbation tests). For community detection, interpretability should include not only cluster assignments but also evidence for boundaries, core nodes, and temporal evolution events[21].

7.6. Efficiency and scalability for streaming and large-scale graphs

Deployments increasingly involve streaming graphs and long time horizons. Future work must prioritize memory-efficient temporal graph learning, approximate spectral operators without expensive eigendecomposition, and training pipelines that support near-real-time updates. Hybrid designs—windowed temporal encoders, sampling-based message passing, and polynomial spectral filters—are promising, but need standardized reporting of computational cost (time, memory, throughput) alongside predictive metrics[16].

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

[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] S. Hochreiter and J. Schmidhuber, “Long Short-Term Memory,” Neural Computation, 1997.

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

[8] A. Kendall and Y. Gal, “What Uncertainties Do We Need in Bayesian Deep Learning for Computer Vision?” NeurIPS, 2017.

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

ISSN: 3079-6342

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

[11] 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.

[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] Y. Ma and D. Qu, “GEL-FMO: Gauge-Equivariant Liquid Fourier-Markov Operators for Uncertainty-Certified Multimodal Reasoning,” IEEE AANN 2025, pp. 604–607, July 2025.

[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] J. D. Loyal and Y. Chen, “A Bayesian Nonparametric Latent Space Approach to Modeling Evolving Communities in Dynamic Networks,” Bayesian Anal., vol. 18, pp. 49–77, 2023.

[20] 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.

[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.

ISSN: 3079-6342

[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.

[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.