A Survey on Decentralized Cyber Defense: Graph Neural Networks, Federated Learning and Blockchain-Based Trust for Intrusion and Lateral Movement Detection

Bhagyshree Patil[1], Bhavana Mekala[2], Tanvi Desai[3], Darshika Mhatre[4], Prof. Kalpana Shewale[5]
[1,2,3,4]*Dept. of Computer Science and Engineering (CSE IoT), A. C. Patil College of Engineering, Kharghar, Navi Mumbai, India*
[5]*Guidance Of Prof A. C. Patil College of Engineering, University of Mumbai*

Abstract— **Modern enterprise and IoT networks face multi-stage attacks, such as lateral movement, that span several hosts and organizations. Conventional machine-learning intrusion detection systems treat network records as independent samples, ignore the relational structure between hosts, and usually require centralizing sensitive logs, which creates privacy and compliance risks. Graph Neural Networks (GNNs) address the first limitation by modelling hosts and their connections as graphs, while Federated Learning (FL) addresses the second by sharing model updates instead of raw data. However, standard FL still trusts the aggregation process, and graph learning, differential privacy and blockchain-based validation are rarely combined in one pipeline. This paper surveys recent work on GNN-based intrusion and lateral movement detection, federated intrusion detection, and blockchain-assisted trust mechanisms, and compares fifteen representative studies by methodology, dataset, reported results and limitations. Six research gaps are identified: centralized graph detectors, flat models in blockchain-based federated systems, integrity-only update validation, privacy–utility trade-offs, federated temporal graphs, and evaluation realism. Finally, a proposed decentralized architecture is outlined that combines graph construction from network and authentication logs, GNN-based anomaly detection, lateral-movement analysis, federated averaging across organizations, differential-privacy accounting, hash-based update validation and attack-target prediction.**

Index Terms— **Cyber defense, intrusion detection, graph neural networks, federated learning, blockchain, lateral movement detection, differential privacy**

I. INTRODUCTION

A. Background

Cyber-attacks against enterprises are increasingly multi-stage: an attacker gains a foothold on one machine and then moves laterally through the network, often with stolen credentials, to reach valuable assets [1],[2]. Intrusion detection systems (IDS) are the first line of defense, but signature-based tools miss novel attacks, and classical machine-learning and deep-learning models usually analyze each flow independently. Benchmark datasets such as UNSW-NB15 [3] are widely used to evaluate such systems, and recent surveys cover federated intrusion detection [4],[5] and its combination with blockchain [6].

B. Challenges in Modern Networks

The growth of IoT and edge devices has enlarged the attack surface and the volume of logs. Organizations cannot easily pool these logs because of privacy, compliance and competitive concerns, and centralized training creates a single point of failure. Federated learning is attractive in this setting because data stay with their owners, while blockchain is proposed to add integrity and auditability [6].

C. Graph Neural Networks and Federated Learning

GNNs learn from graph-structured data, so they can capture host-to-host relationships and attack propagation paths that flat models miss; the graph convolutional network [7] is the standard starting point. In Federated Learning, each client trains locally and a server aggregates the updates; FedAvg combines

local stochastic gradient descent with server-side averaging and is robust to unbalanced and non-IID data [8]. FL alone does not show that submitted updates are honest [9], and privacy of the updates can be addressed with differential privacy and a tracked privacy budget [10].

D. Contributions and Organization

- A survey of GNN-based intrusion and lateral movement detection, federated intrusion detection, and blockchain-assisted federated learning.
- A comparison of fifteen representative studies by methodology, dataset, reported results and limitations (Table I).
- Six research gaps and a proposed decentralized architecture that addresses several of them.

Section II reviews the literature, Section III lists the research gaps, Section IV presents the proposed methodology, and Section V concludes with future scope.

II. LITERATURE SURVEY

A. Federated and Privacy-Preserving Intrusion Detection

Federated learning lets organizations train a shared detector without exchanging raw logs. DeepFed [11] applies this idea to industrial cyber-physical systems with a CNN–GRU detector and a Paillier-based protocol that protects parameters in transit; however, the encryption overhead grows with the number of participants and trust in aggregation still rests on the central server. Differential privacy offers a formal alternative: DP-SGD with a moment's accountant tracks cumulative privacy loss under a budget [10]. Marfo et al. [12] add Gaussian-noise (ε,δ)-differential privacy and checkpoint-based fault tolerance to client selection for network anomaly detection and report 94.8% accuracy on UNSW-NB15 and 90.3% on ROAD, noting that accuracy depends on the privacy budget (86% at $\varepsilon = 10$ and 89% at $\varepsilon = 100$ on UNSW-NB15). A parallel line of work shares cyber threat intelligence (CTI) rather than models: Huff et al. [13] use zero-knowledge proofs and BBS+ signatures for anonymous authentication but exchange only static STIX indicators through a central provider; Priv-Share [14] uses blockchain-based trustless delegation, evaluated analytically with an Ethereum proof of concept; and SeCTIS [15] combines swarm learning, smart contracts and zero-knowledge proofs at the cost of protocol complexity.

B. Graph-Based Intrusion and Lateral Movement Detection

GNNs learn from relations between hosts rather than from isolated records. E-GraphSAGE [16] extends GraphSAGE with edge (flow) features and topology for NetFlow-based IoT intrusion detection and reports better results than the state of the art on four benchmark datasets. EvolveGCN [17] adapts GCN parameters over time with an RNN, so it can handle graphs whose node sets change between snapshots. For lateral movement specifically, Bowman et al. [18] build an authentication graph and flag low-probability logins with an unsupervised link predictor. Euler [1] stacks a GNN on a sequence encoder for temporal link prediction and allows the graph layers to be distributed across machines. Jbeil [2] moves to continuous-time, inductive learning with threat-sample augmentation and reports an AUC of 99.73% and a recall of 99.25% on LANL authentication data even when 30% of the nodes are unseen during training. LMDetect [19] reframes detection as time-aware subgraph classification with local and graph-transformer attention (F1 of 99.03% on LANL and 98.43% on CERT), and UGEA-LMD [20] targets sparse samples and cold-start nodes on continuous-time graphs (AUC of 0.9254 on LANL in the transductive setting). In the works reviewed here, these detectors are trained within a single network.

C. Blockchain-Assisted Federated Learning

Blockchain adds verifiability to federated training. BlockFL [21] lets miners exchange and verify local updates through consensus, removing the central server, and derives an end-to-end latency model; consensus cost is high, and rewards depend on the number of training samples rather than on the quality of the update. IDS-BF [22] combines blockchain and FL for distributed intrusion detection and adjusts client weights during aggregation according to their contribution. FL-BCID [23] records model updates and anomaly scores through smart contracts on a permissioned ledger, applies clipped and noised gradients, and reports 97.3% accuracy on ToN-IoT and N-BaIoT with 41% lower communication overhead in simulation. Wotaifi [24] records SHA-256

hashes of local LSTM weights on a blockchain before FedAvg and reports 99.96% accuracy on ToN-IoT.

D. Security of Federated Aggregation

Federated aggregation is itself an attack surface. Bagdasaryan et al. [9] show that a model-replacement backdoor can be embedded in the global model under FedAvg, and that secure aggregation prevents the server from inspecting individual updates, which makes anomaly checks on updates difficult. Hash-based records, as in [24], protect updates against tampering after submission, but they cannot tell whether a correctly hashed update is itself poisoned.

E. Comparative Analysis

Table I compares fifteen representative studies. Entries in the Dataset and Results columns are reported as given by the original papers.

Table I
Comparison Of Selected Related Work

Ref.	Author & Year	Methodology / Model	Dataset	Results (as reported)	Limitations / Gaps
[11]	Li et al., 2021	CNN + GRU intrusion model for industrial CPS; FL across CPS owners; Paillier-encrypted parameter exchange	Gas-pipeline CPS data	Accuracy 99.2%, Detection Rate 98.8%	Paillier adds computation and communication overhead as participants grow; trust in aggregation rests on the central server, no tamper-evident record
[16]	Lo et al., 2022	GraphSAGE extended with edge (flow) features and topology for NetFlow-based IoT IDS	Four NIDS benchmark datasets	Outperforms state of the art on key classification metrics (as reported)	Needs rich per-flow features that may not be logged uniformly across organizations; single network, non-federated
[17]	Pareja et al., 2020	GCN parameters evolved by an RNN over time-ordered snapshots; no per-node embeddings, handles changing node sets	Link prediction, edge and node classification tasks	Generally higher performance than related approaches (as reported)	Single-owner snapshot history; no privacy or trust mechanism for cross-organization use
[18]	Bowman et al., 2020	Authentication graph from enterprise logs; unsupervised graph learning with a logistic-regression link predictor flags low-probability login	LANL	Precision 94.5%, False Positive Rate < 0.1%	Single organization; fully unsupervised, cannot use confirmed compromised-node labels
[1]	King & Huang, 2022	Model-agnostic GNN stacked on a sequence encoder (e.g., RNN) for temporal link prediction; GCN layers distributable across machines	LANL, DARPA OpTC	Competitive with or better than state-of-the-art link predictors; outperforms other unsupervised lateral-movement detectors (as reported)	Distribution targets scale, not cross-organization privacy; transductive, benign-only training; no trust layer
[2]	Khoury et al., 2024	Continuous-time encoder with decoder for link prediction on unseen nodes (inductive); threat-sample augmentation	LANL, Pivoting	AUC 99.73%, recall 99.25% on LANL with 30% of nodes unseen	Single-network training; no federated, privacy or trust component

Ref.	Author & Year	Methodology / Model	Dataset	Results (as reported)	Limitations / Gaps
[19]	Zhou et al., 2024	Heterogeneous authentication multigraph; time-aware subgraph classification with local and graph-transformer attention	LANL, CERT insider-threat	F1 99.03% (LANL), 98.43% (CERT)	Single site; no federated or trust mechanism; evaluated on a sampled subset of events
[20]	Liu et al., 2025	Continuous-time dynamic graph with event-level incremental encoding; uncertainty-driven embedding augmentation for sparse samples and cold-start nodes	LANL	AUC 0.9254 (transductive)	Centralized; no privacy or trust component
[21]	Kim et al., 2019	Local updates exchanged and verified on a blockchain through consensus; no central server; end-to-end latency model	Latency analysis; no IDS dataset	Optimal block-generation rate characterized	Consensus latency and cost; generic on-device learning, not security or graph data; rewards by sample count, not update quality
[9]	Bagdasaryan et al., 2020	Model-replacement backdoor attack on FedAvg; secure aggregation hinders anomaly checks on individual updates	Image and word-prediction tasks	Backdoor embedded in the global model, triggered on specific inputs, even under FedAvg	Attack-focused; no blockchain- or validation-based defense
[22]	Dagula et al., 2025	Blockchain + FL; contribution-based dynamic client weights in aggregation; Polkadot-based	ToN-IoT dataset	Accuracy 96.8%, Precision 97.1%	Abstract does not describe graph-structured modelling
[23]	Ali et al., 2025	Lightweight FL-trained IDS; permissioned blockchain with smart contracts recording updates and anomaly scores; DP noise on gradients	ToN-IoT, N-BaIoT	97.3% accuracy; 41% lower communication overhead (as reported)	IIoT focus; lightweight non-graph model; evaluated in simulation
[24]	Wotaifi, 2026	Autoencoder feature reduction; per-client LSTM; SHA-256 hashed weights recorded on a blockchain; FedAvg with verification	ToN-IoT	Accuracy 99.96%, precision 99.99%, recall 99.95%, F1 99.97%	Flat (non-graph) model; hashes show integrity, not benignity, of updates; IoT only
[12]	Marfo et al., 2025	Adaptive client selection with Gaussian-noise (ε,δ)-DP and checkpoint-based fault tolerance	UNSW-NB15, ROAD	Accuracy 94.8% (UNSW-NB15), 90.3% (ROAD)	No graph modelling or blockchain/trust layer; accuracy varies with the privacy budget
[15]	Arikkat et al., 2024	Swarm learning + smart contracts + zero-knowledge proofs; validators score data/model quality and participant trust	MISP Cyber Threat Intelligence benchmark	Trust score accuracy 95.4% under Byzantine attacks	Protocol complexity and per-round cost; generic CTI quality scoring, not graph-based lateral movement

III. IDENTIFIED RESEARCH GAPS

G1 — Lateral-movement detectors are centralized. Euler [1], Jbeil [2], LMDetect [19] and UGEA-LMD [20] all learn from the logs of one network. Organizations exposed to similar attackers cannot jointly improve detection without sharing raw authentication logs.

G2 — Federated blockchain IDS rely on flat models. FL-BCID [23] and the work of Wotaifi [24] use lightweight or LSTM models on tabular features, and the abstract of IDS-BF [22] does not describe graph modelling. Host-to-host relations, which matter for lateral movement, are therefore not exploited.

G3 — Updates are checked for integrity, not for benignity. Hash records [24] and consensus-based verification [21] show that an update was not altered, not that it is harmless. A correctly hashed but poisoned update, such as the model-replacement backdoor of [9], would still be accepted.

G4 — The privacy–utility trade-off is under-studied for graph updates. Differential privacy is evaluated on tabular IDS data [12], where accuracy changes with the privacy budget. Its effect on federated graph models for lateral movement is not reported in the reviewed works, and the formal accounting of [10] is rarely applied in this setting.

G5 — Temporal graph models are not federated. EvolveGCN [17], Euler [1], Jbeil [2] and UGEA-LMD [20] model evolving networks for a single owner, so how evolving parameters can be learned jointly across organizations remains open.

G6 — Evaluation realism and cost. A recent preprint is devoted to fair and realistic performance evaluation of graph-based lateral movement detectors [25], which suggests that current protocols are an open issue. Blockchain overhead is mostly measured in simulation [23].

IV. PROPOSED METHODOLOGY

A. System Overview

The proposed system connects log ingestion, graph construction, anomaly detection, lateral-movement analysis, federated learning, update validation and attack prediction (Fig. 1). Each participating organization keeps its raw network and authentication logs locally; only model updates leave the organization.

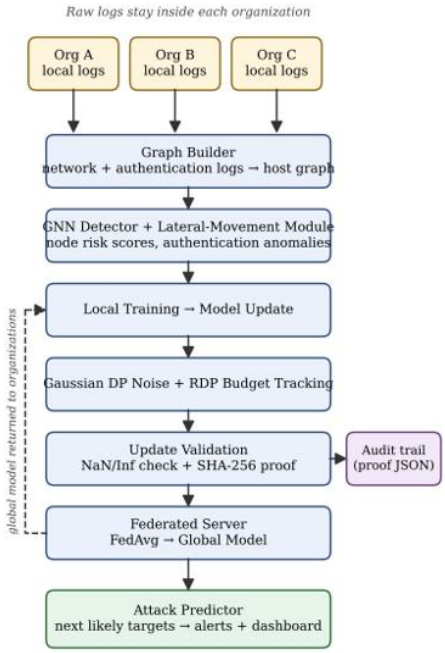

Fig. 1. Proposed decentralized cyber-defense architecture. Raw logs remain inside each organization; only validated, noised model updates are aggregated.

B. Modules

1) Graph construction. Network and authentication logs are converted into host graphs in which hosts are nodes and connections or authentication events are edges. The graphs are converted into a tensor format for GNN training.
2) GNN detector. A GNN assigns a risk score to each node, and nodes above a configurable threshold are marked suspicious. The GNN model can be selected from the dashboard (for example, GAT).
3) Lateral-movement module. Authentication events are ordered into sequences and converted into sequence graphs; the module flags authentication-only anomalies and candidate propagation paths of limited hop count.
4) Federated layer. Each organization trains locally and sends only a model update to the server. The server aggregates by federated averaging once all

expected organizations have submitted, and publishes the global model for clients to pull.

5) Privacy. Gaussian noise is added to each update, and the cumulative privacy budget is tracked against a target (ε, δ) with a Renyi-DP accountant; a conservative approximation is used when the accountant library is unavailable.
6) Update validation. Updates containing NaN or infinite values, or empty updates, are rejected. For accepted updates a deterministic SHA-256 hash is computed and stored as a proof record (hash, organization identifier, timestamp, parameter count and names); proofs are exported as a JSON audit trail and can optionally be recorded on a blockchain.
7) Prediction and alerting. From suspicious nodes, the system predicts the next likely attack targets with confidence scores and raises alerts on a login-protected dashboard with command-center, investigation, privacy and federated views.

C. Prototype Status

The prototype simulates three organizations. Table II separates what is implemented from what is planned; this survey does not claim experimental results for the prototype.

Table II
Prototype Status of Proposed System

Component	Status
Graph construction from network and authentication logs	Implemented in prototype
GNN anomaly detector with selectable GNN model	Implemented in prototype
Lateral-movement module (event sequences to graphs)	Implemented in prototype
Federated training: 3 organizations, FedAvg server	Implemented (simulation)
Gaussian DP noise with RDP-based budget tracking	Implemented (fallback mode if accountant library absent)
SHA-256 proof-of-checksum, NaN/Inf check, JSON audit trail	Implemented
On-chain recording (Ethereum / Ganache)	Experimental
Temporal modelling across graph snapshots	Planned
Robust aggregation against poisoned updates	Planned
Evaluation on public benchmarks with multiple organizations	Planned

D. Relation to the Identified Gaps

Federated training of graph detectors across organizations addresses G1 and, together with the graph model, G2. The RDP-based budget tracking provides the basis for studying G4, although a formal guarantee additionally requires bounded update sensitivity, for example by clipping [10]. Hash-based validation covers integrity only, so G3 is not yet addressed and is listed as future work, as are temporal federated models (G5) and a benchmark evaluation with realistic protocols (G6).

E. Technology and Dataset Overview

The prototype is written in Python. Graphs are handled with NetworkX and PyTorch Geometric, privacy accounting uses the Opacus library, updates are hashed with SHA-256, and the federated server and clients communicate through REST endpoints for update submission, global-model retrieval, status and validation history. Per-organization logs are simulated; public data such as UNSW-NB15 [3] and the LANL authentication data used in [2] are candidates for the planned quantitative evaluation.

V. CONCLUSION AND FUTURE SCOPE

This paper surveyed graph-based intrusion and lateral movement detection, federated intrusion detection and blockchain-assisted federated learning. Graph detectors are accurate on authentication data but are trained within one network; federated and blockchain-based systems add privacy and traceability but mostly use flat models and verify the integrity of updates rather than their benignity. Six research gaps were identified, and a decentralized architecture combining graph learning, federated averaging, differential-privacy accounting and hash-based update validation was proposed to address several of them.

Future work includes: (i) robust aggregation, such as norm clipping and anomaly checks on updates, against poisoning and backdoors [9]; (ii) federated temporal graph models for evolving networks [1],[2],[20]; (iii) on-chain recording with a permissioned ledger and measurement of its overhead [23]; (iv) per-example clipping so that the privacy accounting gives a formal guarantee [10]; and (v) evaluation with realistic protocols on public benchmarks [25].

REFERENCES

[1] King, I. J., and H. H. Huang, "Euler: Detecting network lateral movement via scalable temporal link prediction," in Proc. Network and Distributed System Security Symposium (NDSS), 2022, doi: 10.14722/ndss.2022.24107.

[2] Khoury, J., D. Klisura, H. Zanddizari, G. De La Torre Parra, P. Najafirad, and E. Bou-Harb, "Jbeil: Temporal graph-based inductive learning to infer lateral movement in evolving enterprise networks," in Proc. IEEE Symposium on Security and Privacy (SP), 2024, doi: 10.1109/SP54263.2024.00009.

[3] Moustafa, N., and J. Slay, "UNSW-NB15: A comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set)," in Proc. Military Communications and Information Systems Conference (MilCIS), IEEE, 2015.

[4] Hernandez-Ramos, J. L., G. Karopoulos, E. Chatzoglou, V. Kouliaridis, E. Marmol, A. Gonzalez-Vidal, and G. Kambourakis, "Intrusion detection based on federated learning: A systematic review," ACM Computing Surveys, vol. 57, no. 12, 2025, doi: 10.1145/3731596.

[5] Belenguer, A., J. A. Pascual, and J. Navaridas, "A review of federated learning applications in intrusion detection systems," Computer Networks, vol. 258, 2025, doi: 10.1016/j.comnet.2024.111023.

[6] Ali, S., Q. Li, and A. Yousafzai, "Blockchain and federated learning-based intrusion detection approaches for edge-enabled industrial IoT networks: A survey," Ad Hoc Networks, vol. 152, Jan. 2024.

[7] Kipf, T. N., and M. Welling, "Semi-supervised classification with graph convolutional networks," in Proc. International Conference on Learning Representations (ICLR), 2017, arXiv:1609.02907.

[8] McMahan, H. B., E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, "Communication-efficient learning of deep networks from decentralized data," in Proc. 20th International Conference on Artificial Intelligence and Statistics (AISTATS), PMLR, vol. 54, 2017.

[9] Bagdasaryan, E., A. Veit, Y. Hua, D. Estrin, and V. Shmatikov, "How to backdoor federated learning," in Proc. 23rd International Conference on Artificial Intelligence and Statistics (AISTATS), PMLR, vol. 108, 2020, arXiv:1807.00459.

[10] Abadi, M., A. Chu, I. Goodfellow, H. B. McMahan, I. Mironov, K. Talwar, and L. Zhang, "Deep learning with differential privacy," in Proc. ACM SIGSAC Conference on Computer and Communications Security (CCS), 2016, doi: 10.1145/2976749.2978318.

[11] Li, B., Y. Wu, J. Song, R. Lu, T. Li, and L. Zhao, "DeepFed: Federated deep learning for intrusion detection in industrial cyber-physical systems," IEEE Transactions on Industrial Informatics, 2021, doi: 10.1109/TII.2020.3023430.

[12] Marfo, W., D. K. Tosh, and S. V. Moore, "Adaptive client selection in federated learning: A network anomaly detection use case," arXiv:2501.15038, 2025.

[13] Huff, P., S. Massengale, T. V. X. Phuong, and S. N. G. Gourisetti, "A privacy-preserving cyber threat intelligence sharing system," in Proc. IEEE International Conference on Trust, Privacy and Security in Intelligent Systems and Applications (TPS-ISA), 2024, doi: 10.1109/TPS-ISA62245.2024.00016.

[14] Dunnett, K., S. Pal, Z. Jadidi, V. Dedeoglu, and R. Jurdak, "Priv-Share: A privacy-preserving framework for differential and trustless delegation of cyber threat intelligence using blockchain," Computer Networks, vol. 252, Art. no. 110686, 2024, doi: 10.1016/j.comnet.2024.110686.

[15] Arikkat, D. R., M. Cihangiroglu, M. Conti, R. Rehiman K. A., S. Nicolazzo, A. Nocera, and V. P., "SeCTIS: A framework to secure CTI sharing," arXiv:2406.14102, 2024.

[16] Lo, W. W., S. Layeghy, M. Sarhan, M. Gallagher, and M. Portmann, "E-GraphSAGE: A graph neural network-based intrusion detection system for IoT," in Proc. IEEE/IFIP Network Operations and Management Symposium (NOMS), 2022, doi: 10.1109/NOMS54207.2022.9789878.

[17] Pareja, A., G. Domeniconi, J. Chen, T. Ma, T. Suzumura, H. Kanezashi, T. Kaler, T. B. Schardl, and C. E. Leiserson, "EvolveGCN: Evolving graph convolutional networks for dynamic graphs," in Proc. AAAI Conference on Artificial Intelligence, 2020, arXiv:1902.10191.

[18] Bowman, B., C. Laprade, Y. Ji, and H. H. Huang, "Detecting lateral movement in enterprise computer networks with unsupervised graph AI," in Proc. 23rd International Symposium on Research in Attacks, Intrusions and Defenses (RAID), USENIX Association, 2020.

[19] Zhou, J., J. Yao, X. Chen, S. Yu, Q. Xuan, and X. Yang, "Lateral movement detection via time-aware subgraph classification on authentication logs," arXiv:2411.10279, Nov. 2024.

[20] Liu, J., Y. Shao, S. Zhang, F. Shan, and J. Li, "UGEA-LMD: A continuous-time dynamic graph representation enhancement framework for lateral movement detection," Computers, Materials & Continua, vol. 86, no. 1, pp. 1–20, 2026, doi: 10.32604/cmc.2025.068998.

[21] Kim, H., J. Park, M. Bennis, and S.-L. Kim, "Blockchained on-device federated learning," IEEE Communications Letters, 2019, arXiv:1808.03949.

[22] Dagula, L., L. Xu, K. Gai, and L. Zhu, "A distributed intrusion detection system based on blockchain and federated learning," IEEE, 2025.

[23] Ali, A., et al., "Federated learning-enhanced blockchain framework for privacy-preserving intrusion detection in industrial IoT," arXiv:2505.15376, 2025.

[24] Wotaifi, T. A., "A blockchain-integrated federated learning model and autoencoder-based feature reduction for improving IoT intrusion detection," Journal of Intelligent Informatics, Networks, and Cybersecurity, vol. 2, no. 2, Art. no. 2, 2026, doi: 10.65445/3106-1192.1013.

[25] "On fair and realistic performance evaluations for graph-based lateral movement detectors," arXiv:2607.29390, 2026.