Street Art & Urban Creativity | Vol. 11, No. 1, 2025 | ISSN 2183-9956
Street Art & Urban Creativity. Scientific Journal

Self-Adaptive Cybersecurity Intelligence Using Graph Neural Networks and AI Agents for Critical Infrastructure Protection

CHAITANYA NAGABHYRU[1]
[1]12-1006, RAHMATH NAGAR 2ND LINE, CHILAKALURIPET, GUNTUR PIN: 522616, ANDHRA PRADESH, INDIA
CHAITANYANAGABHYRU@GMAIL.COM

Abstract

Hospitals, water systems, electric utilities, transit agencies and public-sector networks are attacked through chains of ordinary-looking relationships: a stolen credential logs into a workstation, the workstation reaches a server, the server talks to a controller. Detectors that score each device or log line in isolation miss these chains, and automated responses that act without oversight can disrupt the very services they protect. This paper presents a defensive framework that represents users, hosts, operational-technology (OT) devices and external endpoints as a heterogeneous time-windowed graph, classifies known attack behavior with a relational graph neural network, flags previously unseen behavior with a reconstruction-based novelty module, explains each alert by the relationships that drive it, and recommends risk-tiered response actions that require human approval when they could affect critical assets. Because operational security telemetry cannot be released, the framework is evaluated on a controlled synthetic benchmark of 530 entities over 30 days, with three attack families seen in training and two held out, repeated over three random simulations. At a calibrated false-positive rate of 0.20% per node-window, the framework detects 81.2 ± 9.8% of malicious node-windows and every campaign, with a mean latency of 4.3 hours. On unseen families it reaches 45.0% against 18.6% for a gradient-boosting baseline with neighbor features. The top-3 evidence edges are attack-related 66.3% of the time, against 26.7% for random selection. Adversarial hardening keeps detection at 82.4% under half-strength behavior mimicry, where the undefended models fall to 58.5%. The results are simulation-based and indicate behavior of the method, not field performance.

Keywords: Graph Neural Networks, Cybersecurity, Critical Infrastructure, Novelty Detection, Explainable Alerts, Adversarial Robustness, Human-in-the-Loop Response, AI Agents.

1. Introduction

Critical infrastructure is a prime target because disruption is visible, costly and politically meaningful. The 2015 attack on Ukrainian distribution utilities combined stolen credentials, remote access to operator workstations and malicious commands to substation equipment [5]. A 2024 joint advisory by U.S. agencies described state-sponsored actors maintaining persistent, low-profile access in U.S. critical infrastructure networks, with living-off-the-land techniques that blend into normal administration [4]. Operational technology adds a further difficulty: guidance on securing industrial systems stresses that availability and safety constraints limit how aggressively defenders may probe, patch or isolate devices [3].

Two properties of such attacks make conventional detection hard. First, the evidence is relational. A failed-login burst on one host, an unusual process on another and a rare outbound connection from a third may each look benign, yet together they trace a path. Graph representations of authentication, network and command events have therefore attracted growing interest, and graph neural networks (GNNs) can combine a node's own behavior with that of its neighbors [6], [7], [8]. Second, defenders must detect behavior they have not seen before. Supervised classifiers trained on historical incidents do well on repeats of those incidents and poorly on novel tactics, while unsupervised anomaly detectors catch novelty only at the cost of many false alarms [22].

Operators also need alerts they can act on. An alert that states only a risk score invites either blind trust or dismissal. In a hospital or a water utility, the cost of a wrong automated action, such as isolating a controller or disabling a clinician's account, can exceed the cost of the intrusion it targets. Human-factors research on automation has long shown that over-reliance and disuse both arise when operators cannot see why a system acts [25]. Trustworthy deployment therefore requires explanations and an explicit approval step for consequential actions, an emphasis shared by NIST's AI Risk Management Framework and by the ongoing work on trustworthy AI for critical infrastructure [1], [2].

This paper develops and evaluates a framework that addresses these needs together. The objectives are: (i) to detect suspicious relationships among devices, users, network events and attack patterns using a relational GNN; (ii) to flag previously unseen behavior without labels for it; (iii) to explain each alert by the specific relationships that drive it; (iv) to recommend response actions whose impact is matched to asset criticality and gated by human approval; and (v) to quantify resilience when an adversary tries to evade detection. Performance is measured by detection rate, false-positive rate, detection latency and resilience against adversarial attacks.

The setting is one of national interest, since public services, local utilities and hospitals depend on networks that are rarely staffed like a large enterprise security operations center. Realistic labeled telemetry from such systems is not public, so all experiments use a transparent synthetic benchmark whose generating process is described in Section V; results should be read as properties of the method on that benchmark. The paper is organized as follows: Section II reviews the literature, Section III positions related work, Section IV presents the methodology, Section V the experimental setup, Section VI the results, Section VII the discussion and limitations, and Section VIII the conclusions.

2. LITERATURE REVIEW

Machine learning for intrusion detection has a long record of strong laboratory results and weaker deployments. A widely cited critique argues that anomaly detection in networks is harder than in other domains because normal traffic is diverse, errors are costly, and evaluations rarely reflect operational base rates [22]. More recent guidance on security machine learning catalogues recurring pitfalls, including data leakage between training and test periods, unrealistic class balance and evaluation at thresholds that cannot be reproduced in practice [23]. These lessons shape the protocol used here: strictly separated time periods, thresholds fixed on a validation period with a stated false-positive budget, and explicit reporting of campaigns that no method detects.

Graph methods for security build on advances in representation learning. Graph convolutional networks aggregate neighbor features through a normalized adjacency matrix [6]. Inductive variants such as GraphSAGE learn aggregation functions that generalize to unseen nodes [7], and relational GCNs add a separate transformation for each edge type, which suits heterogeneous data where an authentication edge and a network-flow edge carry different meanings [8]. In security, graph models have been applied to detect lateral movement from authentication graphs, using unsupervised link prediction over temporal snapshots [9], [10], and to provenance graphs for detecting advanced persistent threats [20]. Public corpora of authentication and network events from enterprise environments have supported much of this research [11].

Detecting previously unseen behavior is a distinct problem. One-class and reconstruction-based methods model normal behavior and treat departures as suspicious. Deep one-class classification trains a network to map normal data into a compact region [14], and isolation forests score points by how easily they are separated from the rest [13]. Surveys group anomaly detectors by assumption and note that no single family dominates across data types [12]. The practical difficulty is the operating point: a detector with good ranking can still be unusable if it cannot hold false alarms to a rate that analysts can handle.

Explainability for graph models includes post-hoc methods that identify a small subgraph and set of features sufficient to reproduce a prediction [19]. In security, an explanation is useful if it points to the connections and events an analyst can verify, and if it can be checked against ground truth, which is possible only in simulation. Adversarial robustness has been studied for graph learners: small changes to edges or node features can flip predictions [15], [16], echoing earlier findings on perturbation of image classifiers [17]. Adversarial training, in which the model is trained on perturbed inputs, is a standard defense [18], but the strength of the

guarantee depends on whether the training perturbations resemble those the attacker will use.

Finally, response automation raises questions of governance. The MITRE ATT&CK knowledge base organizes adversary tactics and techniques and is widely used to label detections and plan responses [24]. NIST's Cybersecurity Framework 2.0 organizes outcomes into govern, identify, protect, detect, respond and recover [2], and the AI Risk Management Framework asks developers to govern, map, measure and manage AI-specific risks [1]. Together they imply that a detection system should document its limits, measure its performance and keep humans accountable for consequential actions.

3. RELATED WORK

Three lines of work are closest to this study. Temporal graph link-prediction models, exemplified by Euler, learn what normal authentication edges look like and flag improbable new edges, achieving scalability by separating a per-snapshot graph encoder from a recurrent temporal model [9]. Unsupervised graph methods for lateral movement have used similar ideas on large enterprise authentication logs [10]. These approaches are well suited to a single behavior class and need no attack labels, but they do not combine labeled knowledge of known tactics with novelty detection, and they do not address explanation or response.

Provenance-based detectors build graphs of system-level events and learn evolving models of normal execution to identify advanced persistent threats [20]. They operate at host granularity, with rich event detail, and are complementary to the entity-level graph used here. Work on graph adversarial examples demonstrates that structural perturbations can degrade GNN predictions [15], [16], but the attack settings are typically citation or social networks in which an attacker can add arbitrary edges, rather than a security graph in which the attacker controls a few hosts and must keep attack traffic functional.

TABLE 1. Positioning of the proposed framework relative to representative prior approaches

Capability	Typical treatment in prior work	Treatment in this framework
Relational evidence	Per-entity features or single edge type	Relational GNN over authentication, flow and command edges in both directions
Known attacks	Supervised classifiers on node features	Supervised head on graph embeddings
Unseen attacks	Separate unsupervised detector	Reconstruction head trained jointly on benign nodes
Explanation	Risk score only, or generic attribution	Edge-occlusion evidence checked against ground truth
Response	Fully automatic or left to the analyst	Risk-tiered actions with human approval gate
Adversarial resilience	Rarely evaluated	Camouflage and mimicry sweeps, with hardened variant

The contribution is the integration and joint evaluation of these components under one protocol, with attention to the operating point, to campaign-level outcomes and to the interaction between detection, explanation and response. No component is claimed to be new on its own. A limitation common to this literature, and shared here, is that realistic labeled data are scarce, so conclusions about deployment must be drawn cautiously

4. METHODOLOGY

Framework Overview

Figure 1 shows the five modules. Telemetry from authentication logs, network flows, OT command logs and host process events is assembled into a heterogeneous graph for each six-hour window (Module 1). A relational GNN produces node embeddings and a probability that each entity is attacker-controlled (Module 2). A reconstruction head on the same embeddings scores how far each entity's behavior lies from the benign manifold (Module 3). For each alert, an explainer ranks the entity's incident relationships by their contribution to the alert and attributes the behavior to the nearest known family or labels it unrecognized (Module 4). A response recommender maps the alert, its evidence and the asset's criticality to a ranked set of actions, executes low-impact reversible actions automatically and routes high-impact actions to an analyst for approval (Module 5). Analyst decisions are logged for audit and later retraining.

.

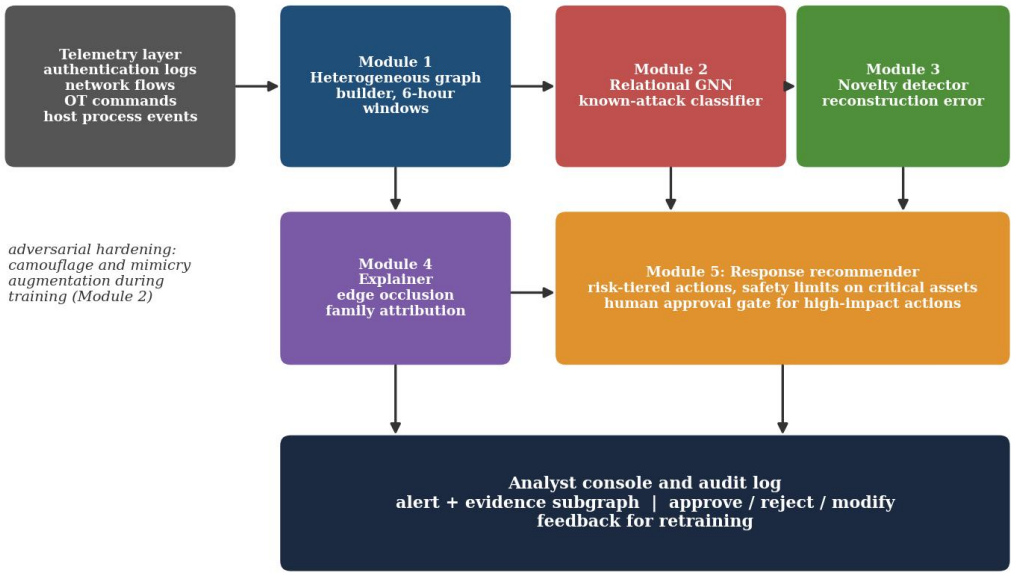


Figure 1: Architecture of the proposed graph-based detection, explanation and human-approved response framework.

Graph Construction and Features

Each window t yields a graph $G_t = (V, E_t)$ whose nodes are users, workstations, servers, OT devices and external endpoints and whose edges are events of three types: authentications, network flows and OT commands. Each type is used in two directions, giving R = 6 relations. Node features summarize the window: counts of authentications initiated and received, failed-authentication ratio, flow counts, log bytes in and out, number of distinct destinations, the fraction of destinations never contacted before, night-time activity, a beacon-periodicity score for outbound flows, the share of flows to rarely seen external endpoints, OT write commands sent and received, command volume, process-spawn counts and privileged-action counts. Static features are a node-type indicator and an asset-criticality score.

Relational Graph Neural Network

The classifier is a two-layer relational GCN [8] with mean aggregation per relation. With H0=Xand Arthe row-normalized adjacency of relation r, layer lcomputes:

$$H^{l+1} = \mathrm{ReLU}\left(H^l W_0^l + \sum_{r=1}^{R} \widehat{A}_r H^l W_r^l + b^l\right) \qquad (1)$$

A linear output layer on the final embedding H2gives the logit of being attacker-controlled. The network has 32 hidden units, is trained with Adam [21] and a class-weighted logistic loss (positive weight 15), and is implemented directly in NumPy with analytic gradients; the gradients were verified against finite differences. Training uses the first 60 windows only.

Novelty Detection

To detect behavior outside the training attack families, the final embedding is also passed through a six-dimensional bottleneck and a linear decoder that reconstructs the node's own input features. This head is trained on benign nodes only, jointly with the classifier:

$$L = L_{\mathrm{CE}}(y, \sigma(h_v w_o)) + \mu \cdot \frac{1}{|B|} \sum_{v \in B} \|x_v - \hat{x}_v\|^2 \qquad (2)$$

where Bis the set of benign monitored nodes and μ=0.5. The reconstruction error of a node, which includes information from its neighborhood through the embedding, serves as the novelty score. Because the decoder has learned to reproduce benign behavior through a narrow

bottleneck, behavior that departs from the benign manifold is reconstructed poorly.

Alert Decision Rule

Two thresholds are set on the validation period (windows 60–79) from benign monitored nodes only: the supervised logit threshold τsand the novelty threshold τn, each at the 1−0.001quantile. An alert is raised if either score exceeds its threshold, so the nominal false-positive budget is 0.2% per node-window. Nodes that are targets of attack events but not attacker-controlled (victims) are excluded from false-positive accounting, because alerts on them are operationally useful. For comparison, each baseline is calibrated to a false-positive budget of 0.2% on the same validation benign nodes.

$$\text{alert}(v,t) = \mathbf{1}[\ell_{vt} > \tau_s] \vee \mathbf{1}[\varepsilon_{vt} > \tau_n] \qquad (3)$$

Explanation by Edge Occlusion

For an alerted node vat window t, the explainer measures the contribution of each incident event eas the drop in a combined alert margin when that event is removed and the window is re-featurized:

$$\text{alert}(v,t) = \mathbf{1}[\ell_{vt} > \tau_s] \vee \mathbf{1}[\varepsilon_{vt} > \tau_n] \qquad (3)$$

The top three events by Ieform the evidence set. Because the simulator labels which events were generated by the attacker, the evidence can be scored by precision at three, by the share of cases where the top-ranked event is attacker-generated, and by fidelity, defined as the share of alerts that no longer fire when the three events are removed together. Two simple heuristics serve as baselines: the three events with the most bytes and the three with the highest counts, along with three random events. The explainer also compares the node's standardized feature deviations with centroids of known families and reports the nearest family only when the supervised head fired; otherwise the alert is marked as an unrecognized pattern.

Adversarial Hardening

Two evasion strategies are considered. In camouflage, the attacker adds kbenign-looking connections per window from each compromised node to destinations it normally contacts, diluting ratios and neighbor aggregates. In mimicry, the attacker throttles its behavior so that a controllable subset of features (failed-authentication ratio, flow counts, bytes, distinct destinations, new-destination ratio, periodicity, OT writes, process spawns and privileged actions) moves toward the benign mean by a fraction λ:

$$x_j^{'} = x_j - \lambda\left(x_j - \bar{x}_j^{\text{benign}}\right),\ j \in \text{controllable set} \qquad (5)$$

The hardened model is trained on the original training windows plus two perturbed copies with randomly drawn k∈05102040and λ∈00.6. To prevent the model from learning that extra edges are themselves a sign of compromise, camouflage edges are also added to a random 10% of benign nodes in the perturbed copies. Attack edges themselves, and any features outside the controllable set, are not altered.

Human-Approved Response Recommendation

Actions are grouped by impact. Low-impact, reversible actions (step-up authentication, rate limiting, enhanced logging) are executed automatically for any alert. High-impact actions (isolating a host, disabling an account, blocking an external endpoint) are proposed only when the alert margin is at least 2.0, or at least 1.0 for assets with criticality of 0.5 or higher, and require analyst approval before execution. Assets that control physical processes are never isolated automatically. The recommender attaches the evidence set and the unrecognized-pattern flag to the proposal so that the analyst can verify it. Policy outcomes are measured by residual attacker exposure, defined as the share of malicious node-windows not neutralized by an executed action, by criticality-weighted collateral disruption on benign nodes and by the number of approvals required per day.

5. EXPERIMENTAL SETUP

Synthetic Benchmark

The benchmark models an organization with 150 users, 200 workstations, 40 servers (10 of them SCADA servers), 60 OT devices and 80 external endpoints, 530 entities in total, observed for 120 six-hour windows (30 days) with about 2,800 events per window. Normal behavior follows a fixed role-based topology: users authenticate to their own and a few other workstations and to a few servers; workstations contact servers and popular external services; servers contact peers; OT devices report to SCADA servers, which poll them with read commands. Activity follows a diurnal pattern. The simulator also includes hard negatives that resemble attacks: three administrator accounts that periodically log into more than twenty hosts during patching, backup servers that occasionally send large volumes to a rarely used external endpoint, and sporadic benign connections to new destinations.

Five attack families are simulated (Table 2). Three are used for training and evaluation, and two appear only in the test period. Each campaign has a start window, a duration and a set of attacker-controlled nodes, and the generator records which events the attacker produced. Windows 0–59 form the training period, 60–79 the validation period and 80–119 the test period

TABLE 2. Attack families in the synthetic benchmark

Family	Behavior	Principal signature	Availability
A: credential abuse and lateral movement	A stolen account pivots across new hosts	Failed-auth bursts, new destinations, privilege use	Train, validation and test (4, 1, 2 campaigns)
B: data exfiltration	Staging from servers, then bulk upload to a rare external endpoint	Large outbound bytes, rare-external share	Train, validation and test (4, 1, 2 campaigns)
C: command-and-control beaconing	Compromised workstations beacon to a rare endpoint	High periodicity, many small flows	Train, validation and test (4, 1, 2 campaigns)
D: OT command abuse (unseen)	A workstation and a SCADA server issue write commands to OT devices	OT write counts, new command edges, low volume	Test only (4 campaigns)
E: low-and-slow lateral spread (unseen)	A server contacts one or two new hosts per window for 14–22 windows	Small new-destination and privilege increments	Test only (4 campaigns)

.

In the first simulation, the training, validation and test periods contain 517, 700 and 273 attacker-controlled node-windows (1.9%, 7.8% and 1.5% of monitored node-windows). Each test period contains 14 campaigns. The full pipeline, including topology, campaigns and model training, was repeated for three simulation seeds, and results are reported as mean ± standard deviation across seeds, except for the explanation, source-of-detection and response analyses, which use the first seed.

Baselines and Metrics

Baselines are logistic regression and gradient boosting [26] on node features, gradient boosting on node features concatenated with the mean features of neighbors (a strong non-neural graph baseline), and an isolation forest [13] trained on benign nodes. Scikit-learn [27] provides the baselines. The supervised GNN without the novelty module and the novelty module alone are reported as ablations. Detection rate is the share of attacker-controlled node-windows that raise an alert. The false-positive rate is the share of benign, non-victim node-windows that raise an alert. A campaign is detected if any of its attacker-controlled node-windows alerts, and detection latency is the time from the first attacker-controlled window to the first alert, in hours; with six-hour windows, latency is quantized to multiples of six, and it is averaged over detected campaigns only. Ranking quality is reported as the area under the ROC curve and as the detection rate at a false-positive rate of 0.2% determined on the test benign nodes, which compares scores independently of calibration drift.

TABLE 3. Principal experimental settings

Component	Setting
Entities / windows	530 entities; 120 six-hour windows; 6 relations
Train / validation / test windows	0–59 / 60–79 / 80–119
GNN	2 relational layers, 32 hidden units, Adam, 40 epochs, positive weight 15
Novelty head	6-dimensional bottleneck, $\mu = 0.5$, benign nodes only
Operating point	0.1% FPR per module on validation benign nodes (0.2% nominal)
Camouflage budgets k	0, 5, 10, 20, 40 extra connections per attacker node-window
Mimicry strengths λ	0, 0.25, 0.5, 0.75 on the controllable feature set
Hardening	Two perturbed copies of training windows; $k \in \{0,5,10,20,40\}$, $\lambda \in [0, 0.6]$; camouflage on 10% of benign nodes
Repetitions	3 simulation seeds, each with new topology, campaigns and model initialization

Response Simulation

Response outcomes are evaluated by replaying the test-period alerts under three policies. In detect only, no action is taken. In fully autonomous, every alert triggers isolation of the node in the next window, lasting four windows if the alert was false. In tiered with human approval, a low-impact action runs automatically on every alert and neutralizes a true attacker node for four windows with probability 0.4, while an isolation proposal, issued under the rule of Section IV, is approved with probability 0.95 if the alert is true and 0.2 if it is false and executes two windows after the alert. These analyst-behavior parameters are assumptions, chosen to represent an analyst who is helped by the evidence view but is not infallible, and the response results are therefore illustrative rather than empirical.

6. RESULTS

Detection Rate and False-Positive Rate

Table 4 reports detection at the validation-calibrated operating point. On known families, every supervised method detects about 98% of attacker-controlled node-windows, so graph structure adds nothing measurable there: the proposed framework reaches 98.3% and gradient boosting with neighbor features 98.6%. Differences appear on unseen families. The supervised GNN alone detects 28.9% of unseen-family node-windows, comparable to or above the supervised baselines (15.6% to 18.9%), and adding the novelty module raises this to 45.0%. The hardened model reaches 52.1%. Overall detection is 81.2 ± 9.8% for the proposed framework against 74.0 ± 10.4% for the strongest baseline. Detection varies considerably across simulations (the proposed framework scored 67.4%, 86.6% and 89.6% for the three seeds) because the number of unseen-family node-windows differs, and the paired improvement over the baselines is more consistent than the absolute level.

TABLE 4. Detection performance on the test period at the validation-calibrated operating point (mean ± sd over 3 simulations)

Method	Detection rate (%)	Known families (%)	Unseen families (%)	False-positive rate (%)	Campaigns detected (%)	Mean latency (h)
Logistic regression	72.5 ± 12.6	98.7 ± 1.4	15.6 ± 11.2	0.20 ± 0.01	78.6 ± 17.5	9.6 ± 5.3
Gradient boosting (node)	73.9 ± 10.9	98.4 ± 0.8	18.9 ± 13.9	0.22 ± 0.06	81.0 ± 6.7	13.7 ± 3.7
Gradient boosting (+neighbors)	74.0 ± 10.4	98.6 ± 0.9	18.6 ± 12.3	0.26 ± 0.06	81.0 ± 8.9	9.6 ± 6.4
Isolation Forest	42.6 ± 15.7	53.8 ± 16.7	16.0 ± 9.8	0.19 ± 0.01	59.5 ± 8.9	11.5 ± 4.6
GNN (supervised only)	75.1 ± 13.9	98.3 ± 1.2	28.9 ± 16.5	0.10 ± 0.02	90.5 ± 13.5	6.1 ± 3.8
GNN novelty only	8.1 ± 2.8	1.0 ± 0.3	23.7 ± 0.7	0.10 ± 0.02	38.1 ± 3.4	6.2 ± 2.7
Proposed (GNN + novelty)	81.2 ± 9.8	98.3 ± 1.2	45.0 ± 11.0	0.20 ± 0.01	100.0 ± 0.0	4.3 ± 2.9
Hardened GNN + novelty	83.8 ± 6.9	98.3 ± 1.2	52.1 ± 3.9	0.23 ± 0.04	100.0 ± 0.0	1.6 ± 1.9

The achieved false-positive rates show the difficulty of holding a budget. The proposed framework lands at 0.20 ± 0.01%, close to the 0.2% target, whereas gradient boosting with neighbors reaches 0.26 ± 0.06% and the hardened model 0.23 ± 0.04%. Thresholds fixed on a short validation period are noisy. At about 450 monitored entities and four windows per day, 0.2% corresponds to roughly four false alerts per day, a load that an on-call analyst can handle but would not welcome.

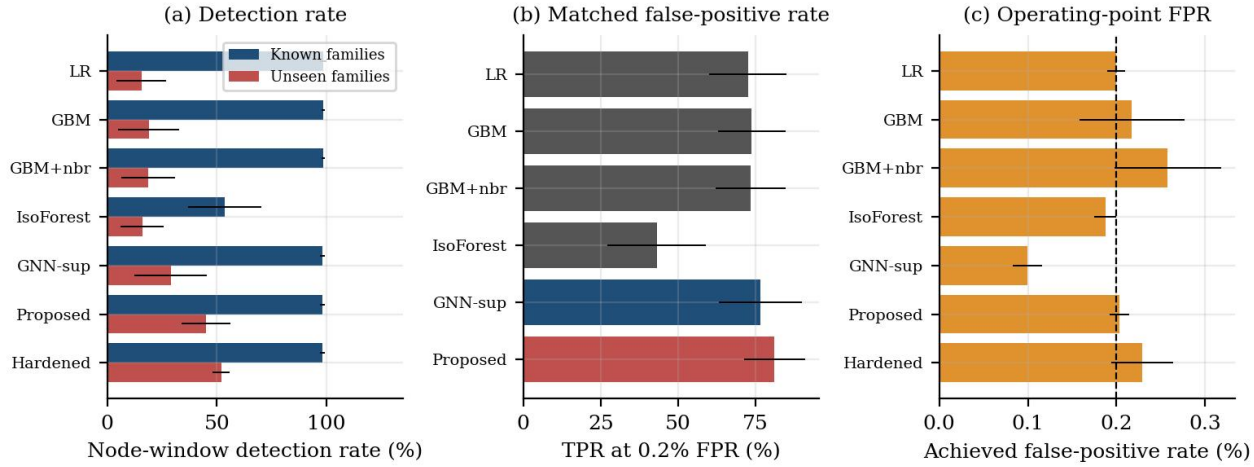


Figure 2: (a) detection rate on known and unseen families; (b) detection rate at 0.2% false-positive rate; (c) achieved false-positive rate at the calibrated operating point

Ranking Quality

Table 5 separates ranking quality from calibration. At a matched 0.2% false-positive rate, the proposed score detects 81.2 ± 9.8% of attacker-controlled node-windows, the supervised GNN 76.6 ± 13.4%, and the three supervised baselines between 72.7% and 73.8%. The isolation forest has a high area under the curve (0.962) but detects only 43.2 ± 15.9% at the matched false-positive rate, which illustrates why area under the curve alone is misleading when the operating region is the extreme low-false-positive tail. The novelty module alone has the highest area under the curve (0.981) yet detects 8.2 ± 2.8% at the matched rate. Its score ranks attackers above typical benign nodes on average, but benign outliers such as patching administrators and backup servers occupy the extreme tail, and the module is useful only in combination with the supervised head.

TABLE 5. Ranking quality on the test period (mean ± sd over 3 simulations)

Method	ROC AUC	Detection rate at 0.2% FPR (%)
Logistic regression	0.917 ± 0.056	72.7 ± 12.5
Gradient boosting (node)	–	73.8 ± 10.9
Gradient boosting (+neighbors)	0.910 ± 0.046	73.4 ± 11.3
Isolation Forest	0.962 ± 0.018	43.2 ± 15.9
GNN (supervised only)	0.882 ± 0.081	76.6 ± 13.4
GNN novelty only	0.981 ± 0.004	8.2 ± 2.8
Proposed (GNN + novelty)	–	81.2 ± 9.8

Which Module Detects What

Table 6 breaks down alerts in the first simulation by the module that fired. For known families, the supervised head alone accounts for 95.4% of attacker-controlled node-windows and 3.3% are missed. For unseen families, the supervised head alone fires on 6.6%, the novelty module alone on 24.6%, and 68.9% are missed. The novelty module thus contributes most of the unseen-family gain, and the majority of unseen-family node-windows still go undetected at this operating point. What saves the framework operationally is persistence: each campaign spans several windows, and the combined detector alerted on at least one node of every test campaign in every simulation.

TABLE 6. Source of detections in the first simulation (share of attacker-controlled node-windows, %)

Family group	Node-windows	Supervised only	Both	Novelty only	Missed
Known (A, B, C)	151	95.4	1.3	0.0	3.3
Unseen (D, E)	122	6.6	0.0	24.6	68.9

Detection Latency and Campaign-Level Outcomes

At campaign level (Figure 3), the proposed framework detected all 14 test campaigns in each simulation, including all four D and all four E campaigns. The baselines detected all A, B and C

campaigns, except the isolation forest, which found only 0% of B and 17% of C campaigns. For the unseen families, logistic regression and both gradient-boosting variants detected 42% of D campaigns and between 83% and 92% of E campaigns. The mean latency, in detected campaigns, was 4.3 ± 2.9 hours for the proposed framework, 6.1 ± 3.8 for the supervised GNN, and 9.6 to 13.7 hours for the baselines. Latency is measured from the first attacker-controlled window with six-hour resolution, so differences below six hours reflect the share of campaigns alerted in the first window and not minute-level timing, and baselines miss more of the hardest campaigns, which are excluded from their latency averages.

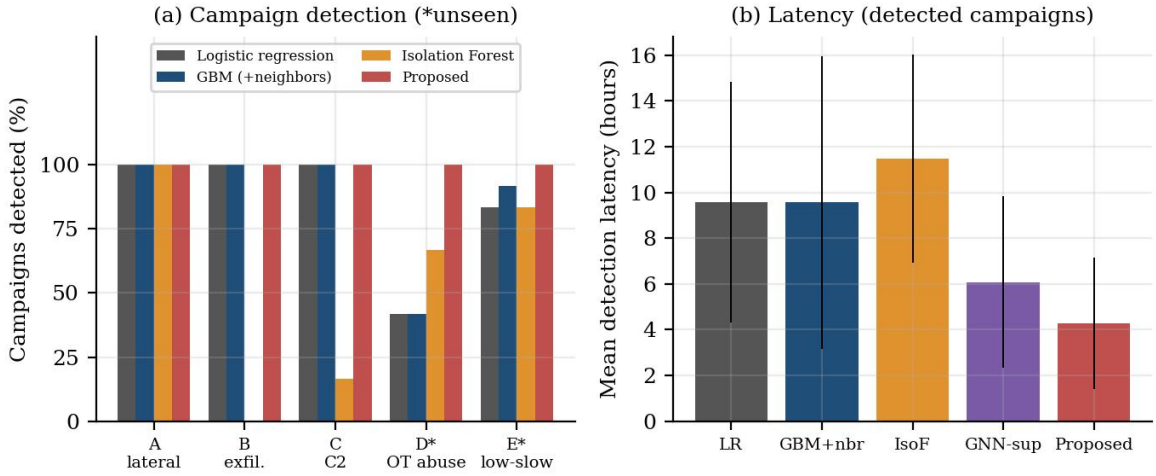


Figure 3: (a) share of campaigns detected, by family (*unseen); (b) mean detection latency over detected campaigns

Explainable Alerts

Of 184 true alerts in the first simulation's test period, 182 had at least four incident events including at least one attacker-generated event and were scored for explanation quality. An alerted node had on average 17.2 incident events in its window, of which 26.4% were attacker-generated. The top-three events chosen by occlusion were attacker-generated 66.3% of the time (precision at three), against 26.7% for random selection, 23.4% for highest count and 10.1% for most bytes. The top-ranked event was attacker-generated in 94.0% of cases (random 27.5%). Removing the three highest-ranked events cleared the alert in 44.5% of cases, against 7.7% for random and 6.0% for top-bytes removal. Precision varied by family, ranging from 33% for exfiltration (only 7 alerts) and 37% for beaconing (few attacker edges among many benign edges, and the signature lies partly in node features rather than edges) to 76% for lateral movement and 91% for OT command abuse, whose command edges are rare.

TABLE 7. Explanation quality for true alerts (first simulation)

Evidence selection	Precision at 3 (%)	Alerts cleared by removal (%)
GNN edge occlusion (proposed)	66.3	44.5
Random incident events	26.7	7.7
Highest-bytes events	10.1	6.0
Highest-count events	23.4	–

Figure 4 shows an evidence subgraph for a lateral-movement alert, and the corresponding analyst-facing text is generated from the same data. The alert concerned a workstation in window 91. Its largest standardized deviations were failed-auth ratio 0.36 (4.4 standard deviations above the training mean), new-destination ratio 0.23 (4.0 standard deviations above the training mean), privilege use 1.39 (3.2 standard deviations above the training mean). The three most influential events were an inbound authentication from a user, an outbound flow to a server and an outbound authentication to another workstation; only the last was generated by the attacker. This example is representative of the imperfection of the explanations: the top three events were not all attack-related, which is why the evidence is presented to the analyst as a ranked list with the supporting numbers, not as a verdict.

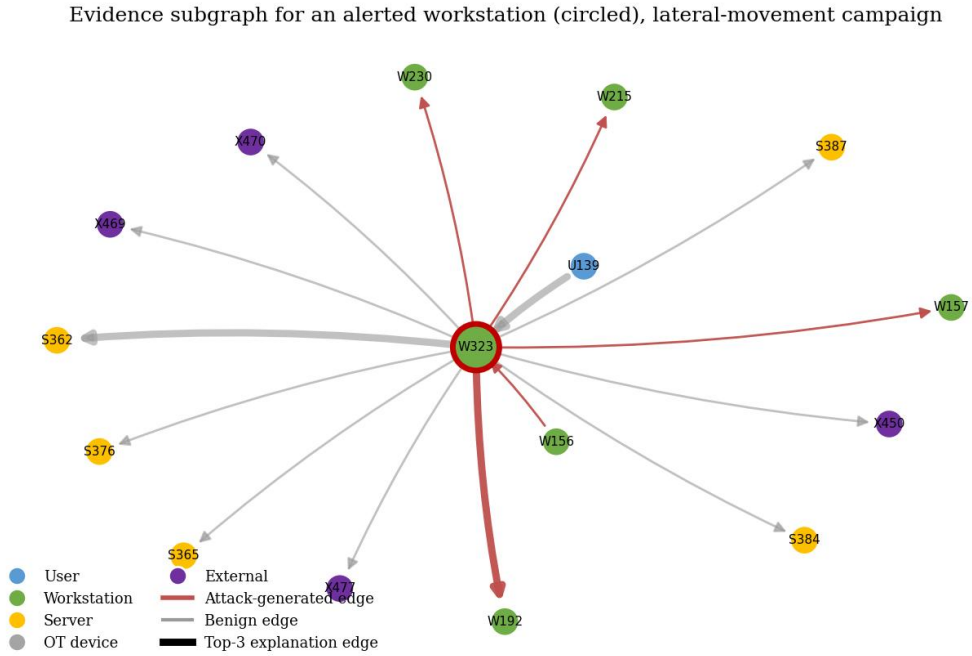


Figure 4: Evidence subgraph for a workstation alert in a lateral-movement campaign; thick edges are the top-3 explanation edges

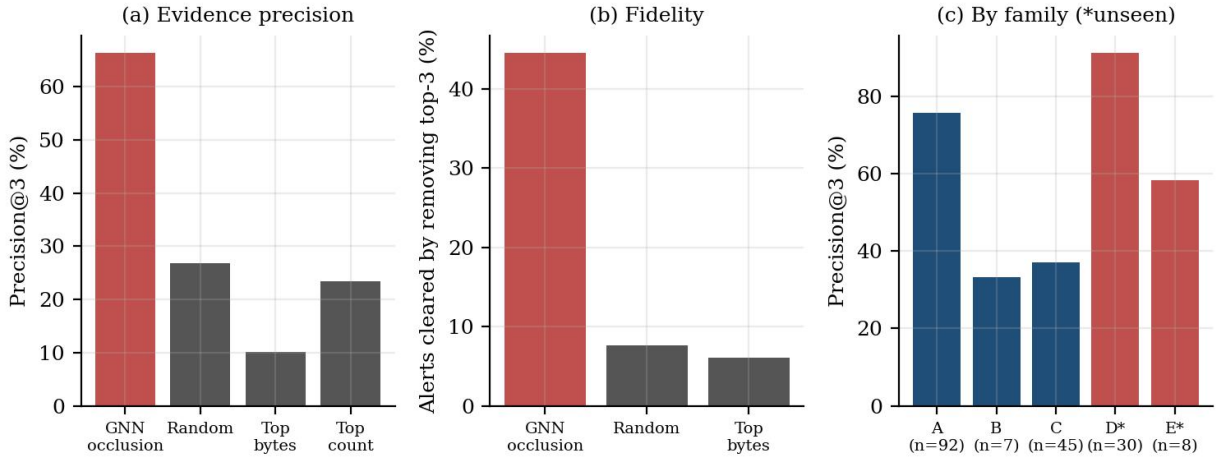


Figure 5: (a) precision of the three highest-ranked evidence events; (b) fidelity of the explanation; (c) precision by attack family (*unseen)

Resilience against Adversarial Attacks

Table 8 and Figure 6 report detection rates when attacker nodes are perturbed at test time. Under camouflage, the proposed framework changes little (from 81.2 ± 9.8% with no perturbation to 76.7 ± 14.7% at 40 extra connections), while the neighbor-feature gradient-boosting baseline falls from 74.0 ± 10.4% to 58.8 ± 16.4% and the isolation forest from 42.6 ± 15.7% to 32.5 ± 15.3%. The hardened model is essentially unaffected (83.8 ± 6.9% to 85.0 ± 5.7%).

Mimicry is far more damaging. At $\lambda = 0.5$, the proposed framework falls from 81.2% to 58.5% and at $\lambda = 0.75$ to 11.7%, and the neighbor-feature baseline falls to 21.9% and 3.4%. The hardened model holds 82.4% at $\lambda = 0.5$ and 75.1% at $\lambda = 0.75$, although its training perturbations stopped at $\lambda = 0.6$ and used the same threat model as the test, which makes this an in-distribution result. The false-positive rate of the proposed framework was unchanged across perturbations (between 0.20% and 0.22%), since only attacker nodes were modified. The attacker in this evaluation is not adaptive to the detector's parameters and cannot alter the attack edges themselves, so the figures are a lower bound on the harm a motivated adversary could cause.

TABLE 8. Detection rate (%) of attacker-controlled node-windows under evasion (mean ± sd over 3 simulations)

Condition	Isolation Forest	GBM + neighbors	GNN (supervised)	Proposed	Hardened
Camouflage k = 0	42.6 ± 15.7	74.0 ± 10.4	75.1 ± 13.9	81.2 ± 9.8	83.8 ± 6.9
Camouflage k = 10	42.5 ± 13.7	71.7 ± 11.3	75.8 ± 14.6	81.8 ± 10.7	84.8 ± 6.1
Camouflage k = 40	32.5 ± 15.3	58.8 ± 16.4	70.6 ± 18.6	76.7 ± 14.7	85.0 ± 5.7
Mimicry λ = 0.25	20.4 ± 9.2	67.5 ± 12.4	70.5 ± 13.9	77.0 ± 10.0	83.2 ± 7.3
Mimicry λ = 0.5	5.5 ± 4.6	21.9 ± 4.8	51.0 ± 17.2	58.5 ± 14.8	82.4 ± 7.8
Mimicry λ = 0.75	0.3 ± 0.4	3.4 ± 1.7	7.3 ± 5.3	11.7 ± 7.2	75.1 ± 13.4

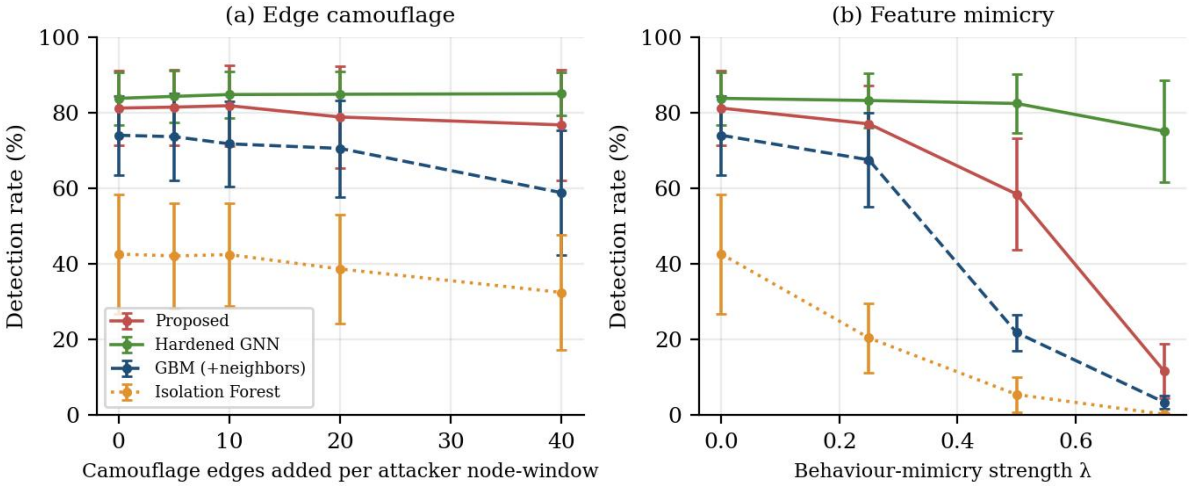


Figure 6: Detection rate under (a) camouflage edges and (b) behavior mimicry for undefended and hardened models

Response Policies

Table 9 and Figure 7 compare the three response policies on the first simulation. With detection only, attackers keep all of their node-windows (100% exposure). Fully autonomous isolation reduces residual exposure to 33.3% but disrupts criticality-weighted benign activity of 86.3 node-window units, because every false alert isolates a node. The tiered policy leaves 45.4% exposure and reduces collateral to 3.9, a 95% reduction, at the price of 12.1 percentage points more exposure and an approval workload of 29.5 requests per day. That workload is high; it reflects a first-simulation test period in which a spreading lateral-movement campaign generates many true alerts, and in practice approvals would be batched by campaign or prioritized by criticality. The comparison depends directly on the assumed analyst parameters of Section V and should be read as a demonstration of the trade-off, not a prediction.

TABLE 9. Response policy outcomes on the test period (first simulation, assumed analyst behavior)

Policy	Residual attacker exposure (%)	Collateral disruption	Approvals per day	Alerts per day
Detect only	100.0	0.0	0.0	34.3
Fully autonomous isolation	33.3	86.3	0.0	34.3
Tiered with human approval	45.4	3.9	29.5	34.3

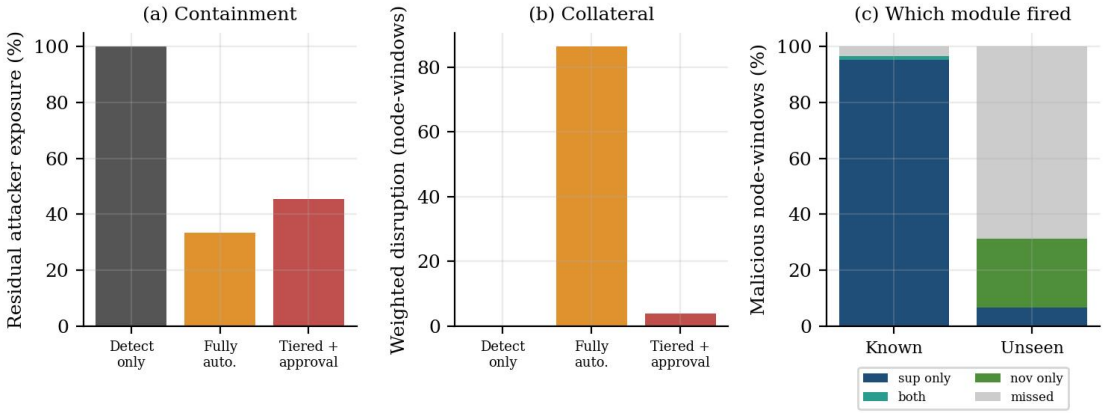


Figure 7: (a) residual attacker exposure and (b) collateral disruption under three response policies; (c) module that fired, by family group

7. DISCUSSION

Four observations follow from the results. First, the main benefit of the framework is on behavior that the classifier was not trained on, and it comes mostly from the combination of a supervised and a novelty signal. On known families, a graph model offered no measurable advantage over gradient boosting with neighbor features in this benchmark, which suggests that when the signature is visible in node features and simple neighbor means, hand-built aggregation is competitive. This is an honest negative result for the claim that graph structure improves known-attack detection, and it should temper expectations on real data in which the benefit of relational modeling may be larger or smaller.

Second, the novelty module alone is not a useful detector at a low false-positive budget, even though its ranking quality is high. Its value is as a second tripwire with its own budget, and the combination was about 4.6 points better than the supervised head alone at a matched false-positive rate. The majority of unseen-family node-windows were still missed. A defender should treat unseen-attack detection as a complement to, and not a replacement for, signature and rule coverage, and should expect to detect such campaigns late and partially, through their most anomalous windows.

Third, adversarial resilience depends on the type of evasion. Adding benign-looking connections is a weak attack against a model that sees both attack edges and node features, but feature mimicry strongly degrades all undefended methods. Hardening by augmentation restored most of the detection, but it was trained and tested against the same family of perturbation. The more general lesson is that the controllable-feature assumption is the key modeling choice for any security-graph robustness claim, and that defenders should assume attackers will learn which features are monitored. Features the attacker cannot change without breaking the attack, such as the existence of command edges to OT devices or the volume of data actually exfiltrated, are more valuable than features that a throttling attacker can adjust.

Fourth, human approval changes the cost structure. Autonomous isolation produced the lowest exposure but also the largest collateral disruption, because every false alert acts on a real asset; tiering cut collateral by roughly 95% in the simulation. The approval workload shows the other side of that trade-off. For small utilities and hospitals with little security staffing, the design question is not whether to keep a human in the loop but which decisions deserve one, and the criticality-aware rule used here is one concrete answer. The approach is consistent with the emphasis of NIST's AI Risk Management Framework on accountability, measurement and transparency [1], and Table 10 maps the framework's controls to its four functions. This mapping is the authors' interpretation and not an official crosswalk.

TABLE 10. Mapping of framework controls to NIST AI RMF functions (authors' interpretation)

AI RMF function	Control in this framework
Govern	Human approval gate for high-impact actions; audit log of alerts, evidence and decisions; asset criticality policy
Map	Defined threat model (Table 2); explicit statement of controllable features and of known and unseen families
Measure	Detection rate, false-positive rate, latency, campaign detection, explanation fidelity, adversarial sweeps across seeds
Manage	Risk-tiered responses; hardening by perturbation; analyst feedback for retraining and threshold review

The limitations are considerable. All data are synthetic and generated by processes the authors designed, so features such as the failed-authentication ratio and periodicity were built to carry the signal the models later find. Only 14 test campaigns per simulation and three simulations were used, so campaign-level percentages are coarse, and detection rates across simulations varied by more than twenty points. Thresholds were fixed on a short validation period, and several methods missed the 0.2% target. The graph model uses independent six-hour snapshots, with no temporal component beyond history-based features. Attack edges, which the attacker cannot hide, make camouflage a weak attack here, while real attackers can also poison training data, which was not examined. The explainer is evaluated only against simulator labels. The response results rest on assumed analyst behavior, a simplified model of neutralization, and no model of attacker adaptation after containment. Finally, the framework has not been tested against real OT protocols, real operational noise or the organizational constraints of a hospital or utility. Future work should validate on shared, de-identified telemetry from partner organizations, add temporal graph models and data-poisoning tests, evaluate adaptive white-box attackers, and study analyst workload with human participants.

8. CONCLUSION

This paper presented a defensive framework that detects suspicious relationships among users, devices and network events with a relational graph neural network, flags previously unseen behavior with a jointly trained novelty head, explains alerts through the relationships that drive them, and recommends risk-tiered responses that require human approval for consequential actions. On a synthetic benchmark with held-out attack families and three repetitions, the framework detected 81.2 ± 9.8% of attacker-controlled node-windows and all campaigns at a false-positive rate of 0.20%, with a mean latency of 4.3 hours. It detected 45.0% of node-windows from unseen families, against 18.6% for the strongest baseline, though most unseen-family node-windows were still missed. Occlusion-based evidence identified attacker-generated events with 66.3% precision at three, behavior mimicry was the dominant evasion threat, and hardening by augmentation preserved detection against it in the tested threat model. Human-approved tiering lowered collateral disruption sharply relative to autonomous isolation at a moderate cost in exposure and a substantial approval workload.

These findings describe the behavior of the method in a controlled setting and are not evidence of field performance. The most valuable next steps are validation on real telemetry, evaluation against adaptive adversaries and studies with analysts, so that claims about trustworthy AI for critical infrastructure rest on operational evidence and not on simulation alone.

References

1. Mahendra Vardhan and S. Sridhar, "Determining False Positive Analysis of Software Vulnerabilities with Predefined Scan Rules using Random Forest Classifier and Decision Tree Technique," 2022 4th International Conference on Advances in Computing, Communication Control and Networking (ICAC3N), Greater Noida, India, 2022, pp. 622-625, DOI: https://doi.org/10.1109/ICAC3N56670.2022.10074458
2. Dip Bharatbhai Pate, Dynamic vs. Static visualizations: Understanding their use cases in big data analysis, International Journal of Science and Research Archive, 2022, 05(02), 392-395., DOI https://doi.org/10.30574/ijsra.2022.5.2.0077

3. Dip Bharatbhai Patel, Building scalable business intelligence systems in the cloud: A technical approach, International Journal of Science and Research Archive, 2021, 02(01), 212-215., Received on 03 February 2021; revised on 17 April 2021; accepted on 19 April 2021,DOI: https://doi.org/10.30574/ijsra.2021.2.1.0047
4. Dip Bharatbhai Patel, Understanding the Impact of Text Analytics on Social Media Sentiment Analysis, Vol. 5 No. 5 (2019) ,September-October, DOI: https://doi.org/10.32628/CSEIT1936413
5. Dip Bharatbhai Patel, Real-Time Data Streaming in BI Dashboards: Opportunities and Challenges, Archives / Vol. 5 No. 4 (2019) : July-August, DOI: https://doi.org/10.32628/CSEIT192142
6. Dip Bharatbhai Patel, the Role of Data Visualization in Modern Intelligence, Archives / Vol. 2 No. 2 / Research Article, Published, 2020-11-28 DOI: https://doi.org/10.32996/jcsts.2020.2.2.4
7. A. Srivastava, "AI-assisted cloud migration of healthcare claims data from legacy systems: A metadata-driven framework," Int. J. Commun. Networks Inf. Secur., vol. 14, no. 3, Nov. 2022. [Online]. Available: https://ijcnis.org/index.php/ijcnis/article/view/8682
8. A. Srivastava, "Enhancing provider and claims data accuracy using artificial intelligence on cloud-native data platforms," J. Comput. Anal. Appl., vol. 31, no. 4, Nov. 2023. [Online]. Available: https://eudoxuspress.com/index.php/pub/article/view/4347/3189
9. A. Srivastava, "Optimizing large-scale data migration for cloud-native architectures," J. Comput. Anal. Appl., vol. 31, no. 2, pp. 652–662, May 2023. [Online]. Available: https://eudoxuspress.com/index.php/pub/article/view/4603
10. Backup Integrity and Recovery Readiness Assessment for High-Availability Databases Raghu Gollapudi, Computer Fraud & Security, Vol. 2023, Issue 8, pp. 35-48, 10.52710/cfs.1031
11. Risk-Controlled Near-Zero-Downtime Oracle Database Migration Using GoldenGate Raghu Gollapudi, International Journal of Computational and Experimental Science and Engineering, 8(3), 2022 ,10.22399/ijcesen.5382
12. Manoj Kumar Kagitha, Automated Security Compliance in Ci/Cd: A Natural Language Processing (NLP) Approach to Detecting Vulnerabilities in Infrastructure-As-Code (Terraform/Bicep),Publication year 2021, Volume 2021, Issue 12, https://computerfraudsecurity.com/index.php/journal/article/view/970
13. Manoj Kumar Kagitha, Self-Healing Cloud Infrastructure: A Hybrid Reinforcement Learning Framework for Predicting Microservice Failures in Azure Kubernetes Service (AKS),Publication year 2023, Volume 2023, Issue 12,https://computerfraudsecurity.com/index.php/journal/article/view/971
14. Manoj Kumar Kagitha, LOAD BALANCING TECHNIQUES IN DISTRIBUTED CLOUD ENVIRONMENTS, (IJESMR - 2017), D.O.I:10.53555/jcr.v6:i1.13380, https://jcreview.com/archives/volume-6/issue-1/13380
15. Manoj Kumar Kagitha, AI-DRIVEN OBSERVABILITY FOR HYPERSCALE COLOCATION DATA CENTERS: A CASE STUDY OF LOG ANALYTICS AND ANOMALY DETECTION PIPELINES, Archives / Vol. 27 No. 7 (2019): JOCAAA,https://www.eudoxuspress.com/index.php/pub/article/view/5053
16. Manoj Kumar Kagitha, GREENOPS IN THE CLOUD: AN AI-DRIVEN TELEMETRY ANALYSIS MODEL FOR REDUCING CARBON FOOTPRINT IN HIGH-AVAILABILITY CLUSTERS, Vol. 28 No. 6 (2020): JOCAAA, https://www.eudoxuspress.com/index.php/pub/article/view/5054
17. Naresh Lokiny. (2022). Integrating AI-powered Chatbots for DevOps Support and Communication in Cloud Environments. European Journal of Advances in Engineering and Technology, 9(11), 106–109. https://doi.org/10.5281/zenodo.13325989
18. Naresh Lokiny, (2021), "Disaster Recovery and Business Continuity Planning in DevOps Cloud with AI", International Journal of Science and Research (IJSR), 10(3), 2024-2027. https://dx.doi.org/10.21275/SR24724151733, https://www.ijsr.net/getabstract.php?paperid=SR24724151733
19. Naresh Lokiny, & Ranganath Nandanampati. (2020). DevSecOps: Integrating Security into DevOps with AI in Cloud. Journal of Scientific and Engineering Research, 7(10), 239–242. https://doi.org/10.5281/zenodo.13348695
20. Naresh Lokiny, & Pradip Reddy. (2021). Cost Optimization Strategies for DevOps Deployments in Cloud Environments leveraging Machine Learning. European Journal of Advances in Engineering and Technology, 8(3), 69–72. https://doi.org/10.5281/zenodo.13325845
21. Jayanth Para, AI Based Cloud Computation Observational Method & Process, Vol. 51 No. 4 (2023): October-December 2023, Power System Protection and Control, ISSN-1674-3415,

https://pspac.info/index.php/dlbh/article/view/171 , DOI: https://doi.org/10.46121/pspc.51.4.3
22. Jobayar Alom, Ahsan Ahmed. (2023). Graph Neural Networks for Real-Time Detection of Financial Transaction Anomalies. Acta Scientiae, 24(5), 82–90. https://doi.org/10.22178/acta.24.5.6
23. Kaleshwar Aryasomayajula, PREVENTING BIAS IN AI-BASED DECISION-MAKING: ANALYZING TECHNIQUES TO REMOVE UNFAIR PREJUDICE IN ALGORITHMIC OUTCOMES, Vol. 50 No. 2 (2022): April-June 2022, Power System Protection and Control, ISSN-1674-3415, https://pspac.info/index.php/dlbh/article/view/294
24. Kaleshwar Aryasomayajula, MODERN DEVELOPMENT PRACTICES: INVESTIGATIONS INTO SERVERLESS COMPUTING, LOW-CODE/NO-CODE PLATFORMS, AND DEVELOPER PRODUCTIVITY IN REMOTE ENVIRONMENTS, Vol. 50 No. 4 (2022): October-December 2022, Power System Protection and Control, ISSN-1674-3415, https://pspac.info/index.php/dlbh/article/view/364
25. Vikas Suresh Bagora, KERNEL-LEVEL PERFORMANCE OPTIMIZATION FOR CARRIER-GRADE BROADBAND AND HIGH-SPEED NETWORKING SYSTEMS, Vol. 47 No. 1 (2019), Power System Protection and Control, ISSN-1674-3415, https://pspac.info/index.php/dlbh/article/view/359
26. Vikas Suresh Bagora, SCALABLE 128G FIBRE CHANNEL AND ETHERNET CONVERGENCE FOR NEXT-GENERATION DATA CENTER NETWORKS, Vol. 50 No. 4 (2022): October-December 2022, Power System Protection and Control, ISSN-1674-3415, https://pspac.info/index.php/dlbh/article/view/362
27. Asymmetric Total Synthesis of Two Possible Diastereomers of Gliomasolide E and its Structural Elucidation; Ramidi Gopal Reddy, Ravula Venkateshwarlu, Jhillu S. Yadav and Debendra K. Mohapatra. (J. Org. Chem. 2017, 82(2), 1053-1063). https://doi.org/10.1021/acs.joc.6b02611 , https://pubs.acs.org/doi/abs/10.1021/acs.joc.6b02611
28. Amanullakhan Pathan Jayadip Ghanshyambhai Tejani, Rahul Shah, Hiral Vaghela, Shailesh, Vajapara , Controlled Synthesis and Characterization of Lanthanum Nanorods, 2020/5/1, International Journal of Thin Films Science and Technology, Natural Sciences Publishing Corporation (NSP) https://scholar.google.com/citations?view_op=view_citation&hl=en&user=efbNMkwAAAAJ & citation_for_view=efbNMkwAAAAJ:M3NEmzRMIkIC
29. Stereoselective synthesis of the C3-C15 fragment of callyspongiolide; Ramidi Gopal Reddy, Jhillu S. Yadav and Debendra K. Mohapatra. (Tetrahedron Letters. 2018, 59, 3579-3582). https://doi.org/10.1016/j.tetlet.2018.08.041, https://www.sciencedirect.com/science/article/pii/S0040403918310426
30. Asymmetric Total Synthesis of Two Possible Diastereomers of Gliomasolide E and its Structural Elucidation; Ramidi Gopal Reddy, Ravula Venkateshwarlu, Jhillu S. Yadav and Debendra K. Mohapatra. (J. Org. Chem. 2017, 82(2), 1053-1063). https://doi.org/10.1021/acs.joc.6b02611, , https://pubs.acs.org/doi/abs/10.1021/acs.joc.6b02611
31. Kaleshwar Aryasomayajula, PREVENTING BIAS IN AI-BASED DECISION-MAKING: ANALYZING TECHNIQUES TO REMOVE UNFAIR PREJUDICE IN ALGORITHMIC OUTCOMES, Vol. 50 No. 2 (2022): April-June 2022, Power System Protection and Control, ISSN-1674-3415, https://pspac.info/index.php/dlbh/article/view/294
32. Kaleshwar Aryasomayajula, MODERN DEVELOPMENT PRACTICES: INVESTIGATIONS INTO SERVERLESS COMPUTING, LOW-CODE/NO-CODE PLATFORMS, AND DEVELOPER PRODUCTIVITY IN REMOTE ENVIRONMENTS, Vol. 50 No. 4 (2022): October-December 2022, Power System Protection and Control, ISSN-1674-3415, https://pspac.info/index.php/dlbh/article/view/364
33. Kaleshwar Aryasomayajula, Micro services: “A Comparative Analysis of Monolithic vs. Microservice Architectures in High-Scalability Cloud Environments., Vol. 51 No. 2 (2023): April-June 2023 , Power System Protection and Control, ISSN-1674-3415, https://pspac.info/index.php/dlbh/article/view/374
34. Controlled Synthesis and Characterization of Lanthanum Nanorods, Author(s), Jayadeep Tejani, Rahul Shah, Hiral Vaghela, Shailesh Vajapara, Amanullakhan Pathan, doi:10.18576/ijtfst/090205, URL: https://www.naturalspublishing.com/Article.asp?ArtcID=20705, May-2020
35. Sudipkumar Ghanvat , Aishwarya Badlani, Shreya Sandeep Joshi, Marketing Effectiveness in the Post-Cookie Era: A Comparative Analysis of First Party and Zero-Party Data Strategies on

Campaign Performance and Customer Equity, Vol. 31 No. 4 (2023): JOCAAA , https://www.eudoxuspress.com/index.php/pub/article/view/3940
36. Chander Vijay S Sanbhi, BAYESIAN UPDATING OF RAM MODELS FOR DYNAMIC AVAILABILITY FORECASTING UNDER REALISTIC DOWNTIME CONSTRAINTS, Vol. 51 No. 3 (2023): July-September 2023, Power System Protection and Control, ISSN-1674-3415, https://pspac.info/index.php/dlbh/article/view/381
37. Chander Vijay S Sanbhi, HYBRID RELIABILITY MODELLING FOR ROTATING EQUIPMENT: PHYSICS- INFORMED LEARNING WITH SPARSE FAILURE DATA, Vol. 51 No. 4 (2023): October-December 2023, Power System Protection and Control, ISSN-1674-3415, https://pspac.info/index.php/dlbh/article/view/382
38. A. Srivastava, "AI-assisted cloud migration of healthcare claims data from legacy systems: A metadata-driven framework," Int. J. Commun. Networks Inf. Secur., vol. 14, no. 3, Nov. 2022. [Online]. Available: https://ijcnis.org/index.php/ijcnis/article/view/8682
39. A. Srivastava, "Enhancing provider and claims data accuracy using artificial intelligence on cloud-native data platforms," J. Comput. Anal. Appl., vol. 31, no. 4, Nov. 2023. [Online]. Available: https://eudoxuspress.com/index.php/pub/article/view/4347/3189
40. A. Srivastava, "Optimizing large-scale data migration for cloud-native architectures," J. Comput. Anal. Appl., vol. 31, no. 2, pp. 652–662, May 2023. [Online]. Available: https://eudoxuspress.com/index.php/pub/article/view/4603
41. Santhosh Chitraju Gopal Varma, The Evolution of Cloud-Native Architectures: Exploring the Synergy between Kubernetes and Microservices, DOI: https://doi.org/10.63282/3050-9246.IJETCSIT-V1I4P104, Vol. 1 No. 4 (2020), https://ijetcsit.org/index.php/ijetcsit/article/view/108
42. Santhosh Chitraju Gopal Varma, Artificial Intelligence in Cloud Computing: Building Intelligent, Distributed, and Fault-Tolerant Systems, Vol. 3 No. 1 (2022), DOI: https://doi.org/10.63282/3050-9416.IJAIBDCMS-V3I1P105, https://ijaibdcms.org/index.php/ijaibdcms/article/view/78
43. Sunil Karmakar, AI driven Identity & Access Management with Oracle Cloud security, Vol. 30 No. 2 (2022): JOCAAA, https://eudoxuspress.com/index.php/pub/article/view/4118
44. Bandari, V. V., & Lekkala, H. B. (2024). Physics-informed reinforcement learning for real-time control of complex manufacturing processes. Power System Protection and Control, 52(2), 164–170. https://pspac.info/index.php/dlbh/article/view/343
45. Bandari, V. V., & Lekkala, H. B. (2024). AI-based operator behavior monitoring and cost optimization using digital traceability in manufacturing systems. Power System Protection and Control, 52(1), 38–45. https://pspac.info/index.php/dlbh/article/view/342
46. Kancharla Vijayalakshmi, Bethapudi Samuel Anand Andrews, Estimation of Cyproheptadine Hydrochloride and Tricholine Citrate Simultaneously in Syrup Dose by Applying Stability Indicating HPLC Methodology, December 31, 2020, Vol. 11, Issue 4, Pages 1813–1819, DOI: 10.26452/ijrps.v11iSPL4.4383, https://ijrps.com/v11si4/3683
47. Vijayalakshmi Kancharla ,Kumar Abbaraju V D N, Andrews B S A, Nagaraju Dasari, Nagu Korupolu, Melphan as Anticancer and New Insights of Drug as Parenteral Dosage form against SARS-CoV-2 by Using Development and Validation , November 18, 2020, Vol. 11, Issue 1, Pages 1437–1446
DOI: 10.26452/ijrps.v11iSPL1.3686, https://ijrps.com/v11si1/2911
48. Ramesh, D., Singh, A., & Channaiah, L. H. (2024). The importance of validated kill steps and thermal death kinetics of Salmonella in food safety management. Global Journal of Nutrition & Food Science, 5(1), Article 000601. https://doi.org/10.33552/GJNFS.2024.05.000601
49. Bharati Reddi, Chandan Kishor,Aruna Jangam, Sandeepchowdary Bala, Uma Rajeswari Batchu ,Rambabu Gundla , Anthony Addlagatta, Regioselectivity in inhibition of peptide deformylase from Haemophilus influenzae by 4- vs 5-azaindole hydroxamic acid derivatives: Biochemical, structural and antimicrobial studies, Date: November 2022, Article: 106095, Volume: Volume 128, 10.1016/j.bioorg.2022.106095, https://www.sciencedirect.com/science/article/abs/pii/S0045206822005016
50. Bhavita Kattula, Bharati Reddi, Aruna Jangam, Lekhika Naik , Bala Manikanta Adimoolam , Suresh Vavilapalli , Sayanna Are , Jagadeshwar Reddy Thota, Surender Singh Jadav , Mohammed Arifuddin Anthony Addlagatta, Development of 2-chloroquinoline based heterocyclic frameworks as dual inhibitors of SARS-CoV-2 MPro and PLPro, Date: 1 July 2023 Article: 124772, Volume: Volume 242, Part 2, 10.1016/j.ijbiomac.2023.124772 https://www.sciencedirect.com/science/article/pii/S0141813023016665

51. Iram Irfan, Asghar Ali, Bharati Reddi, Mohd. Abrar Khan, Phool Hasan, Sarfraz Ahmed Amad Uddin, Magdalena Piatek, Kevin Kavanagh, Mohammad Abid, Design, Synthesis and Mechanistic Studies of Novel Isatin-Pyrazole Hydrazone Conjugates as Selective and Potent Bacterial MetAP Inhibitors, 19 August 2022, https://doi.org/10.3390/antibiotics11081126 https://www.mdpi.com/2079-6382/11/8/1126
52. GAURAV KUMAR SINHA, Efficient Test-Time Compute Scaling for Resource-Constrained Large Language Models Vol. 32 No. 1 (2024): JOCAAA, DOI: https://doi.org/10.1109/ICAC3N56670.2022.10074458 https://eudoxuspress.com/index.php/pub/article/view/5918
53. GAURAV KUMAR SINHA , The Impact of Artificial Intelligence on Predictive Quality Assurance in Modern Manufacturing, Vol. 33 No. 05 (2024): JOCAAA, https://doi.org/10.1109/ICAC3N56670.2022.10074458 , https://eudoxuspress.com/index.php/pub/article/view/5919
54. GAURAV KUMAR SINHA , Hybrid Quantum-Classical Neural Networks for Combinatorial Optimization and Machine Learning, Vol. 33 No. 08 (2024): JOCAAA, DOI: https://doi.org/10.1109/ICAC3N56670.2022.10074458, https://eudoxuspress.com/index.php/pub/article/view/5921
55. Lalith Chandra Bandaru , Mohammed Shakeer Bandrevu, GRAPHSEC: Graph-Based Supply Chain Attack Detection and Risk Propagation Analysis for Enterprise Salesforce Deployment Pipelines, Paper Id: 233135,Published On: 2023-11-03, Published In: Volume 11, Issue 6, November-December 2023, https://www.ijirmps.org/research-paper.php?id=233135 , DOI: https://doi.org/10.37082/IJIRMPS.v11.i6.233135