JCBI
Journal of Computing & Biomedical Informatics
ISSN: 2710 - 1606
Volume 11 Issue 01
2026

Research Article
https://doi.org/10.56979/1101/2026/1491

ShadowMesh: An Execution-Validated Causal Graph Learning Architecture for Detecting Hidden Agent Coalitions, Tracing Malicious Interaction Paths, and Supporting Trustworthy Autonomous Cyber Defense Governance

Mohammed Al-Naeem[1]

[1]Department of Computer Networks and Communications, College of Computer Sciences & Information Technology, King Faisal University, Al Ahsa, 31982, Saudi Arabia.
*Corresponding Author: Mohammed Al-Naeem. Email: naeem@kfu.edu.sa

Received: February 22, 2026 **Accepted:** May 26, 2026

Abstract: Autonomous cyber-defense agents can shorten security-operations-center response time by coordinating telemetry collection, triage, remediation, and reporting. Their collaborative behavior, however, also creates an integrity risk: a compromised or strategically misaligned subgroup can suppress alerts, inject corroborating but misleading evidence, or coordinate unsafe response actions while each individual action appears operationally plausible. This study introduces ShadowMesh, an empirical CybORG-based causal temporal graph framework for detecting and attributing covert collusion in autonomous multi-agent cyber defense networks. The framework was executed across reproducible benign, competitive, and four malicious-coalition behaviors, yielding 2,400 mission episodes and 28,800 temporally aligned graph windows from agent-action sequences, message metadata, tool-call records, and system observations. ShadowMesh uses typed temporal interaction graphs, masked evidence channels for incomplete telemetry, a causal temporal graph transformer, and coalition-level risk scoring to jointly determine whether collusion exists, which coalition pattern is active, which agents are implicated, and which interaction paths support the alert. Episode-disjoint 70/15/15 splits, five controlled random seeds, and stress conditions involving communication delay, missing logs, noisy tool records, heterogeneous agents, and unseen missions were evaluated through actual repeated CybORG executions. The complete model achieved a macro-F1 of 97.2%, a false-positive rate of 1.1%, coalition-attribution accuracy of 96.4%, and a median decision delay of 1.72 s, outperforming static graph, temporal graph, and non-causal variants. Performance remained at or above 93.1% macro-F1 under the reported delay, missing-log, noisy-record, and unseen-mission stress conditions. These results are empirical outcomes of repeated CybORG experiment executions rather than illustrative or fixed numerical arrays; although they are not field-SOC deployment measurements, they demonstrate how causal relational evidence can provide a defensible monitoring layer for autonomous SOC ecosystems where reliable agent accountability is as important as detection accuracy.

Keywords: Autonomous Cyber Defense; Multi-Agent Security; Malicious Coalition Detection; Temporal Graph Learning; Causal Explainability; CybORG; Security Operations Center.

1. Introduction

Security Operations Centers (SOCs) are increasingly evolving from analyst-centered monitoring environments into distributed decision ecosystems in which autonomous components collect telemetry, correlate alerts, enrich incident context, validate evidence, recommend response actions, and, in some cases, initiate containment procedures. This transition is motivated by the scale, speed, and heterogeneity

of modern enterprise infrastructure, where security-relevant evidence is distributed across endpoints, identities, cloud services, network controls, and orchestration platforms. Recent work on autonomous agents, AI-driven cybersecurity, and large language model enabled security workflows indicates that agent specialization can substantially improve response timeliness and operational coverage when evidence collection and reasoning are coordinated across multiple tasks [1]-[4]. Yet the same delegation that makes an agentic SOC effective also creates new trust dependencies: an output produced by one agent can become the input, justification, or action trigger for another agent.

This dependency structure creates an internal security problem that is materially different from conventional intrusion detection. Rather than detecting an external attacker, an SOC may need to determine whether a subgroup of otherwise legitimate defensive agents is coordinating to manipulate the defensive workflow itself. Such a coalition may suppress or delay escalation, mutually reinforce misleading incident claims, conceal tool failures, selectively propagate evidence, or produce compatible but unsafe recommendations. In each case, the behavior of an individual agent can remain operationally plausible when examined in isolation. The harmful signal instead lies in the temporal order, direction, recurrence, and interdependence of actions, messages, tool results, and recommendations. Autonomous cyber-operation benchmarks and multi-agent cyber-defense research provide a useful basis for studying such interactions under partial observability and evolving mission state, but collaborative agents are generally treated as trustworthy by default [19], [20], [29], [30].

Existing security analytics are not designed to expose this form of covert coordination reliably. Conventional anomaly detectors frequently score events, hosts, or network flows independently, while provenance methods often prioritize external attack reconstruction rather than the behavioral integrity of the defensive workflow. Even when graph-based methods are adopted, static graph formulations can obscure the time-sensitive dependencies through which a misleading observation becomes an accepted claim and eventually an operational action. Recent graph-learning and temporal-security studies demonstrate the value of modeling entities, relations, and evolving context, particularly for detecting subtle structured anomalies and high-volume cyber telemetry [5], [8], [21], [22]. However, the central question in this work is more specific: whether the observed collaboration pattern is consistent with benign mission coordination or with a coalition that is covertly manipulating the decision process.

A deployable coalition-monitoring framework must therefore integrate heterogeneous evidence rather than inspect a single message stream. Relevant evidence includes agent action identity and outcome, tool-call provenance, message-routing metadata, host-state observations, confidence assertions, response timing, evidence availability, and the graph structure that links one decision to the next. It must preserve the distinction between an absent record and a negative observation, because missing telemetry can otherwise be mistaken for malicious behavior. It must also remain robust to communication delay, noisy tool records, heterogeneous agent policies, and previously unseen missions while maintaining a sufficiently low false-positive rate for analyst-facing SOC use. These requirements motivate a typed temporal graph representation with role-aware node features, relation-specific edges, explicit time lags, and missing-evidence masks [7], [26], [31], [34].

Interpretability is equally important in this setting. An opaque coalition score is not sufficient when an alert may affect trust in an automated response component or alter the handling of an active incident. Analysts require a reviewable explanation that identifies the participating agents, the interaction paths that increased coalition risk, and the evidence conditions under which the conclusion was reached. ShadowMesh consequently adopts an operational counterfactual notion of causal evidence: the model evaluates how the coalition posterior changes when a claimed decisive action-message-tool dependency is masked while the surrounding temporal context is preserved. This is not a claim of causal identification in an uncontrolled enterprise; it is a controlled consistency test intended to make the attribution more faithful to the model's decision process. Recent work on causal representation learning and counterfactual explanation supports the value of such perturbation-aware analysis for complex graph-based inference [23], [32].

The literature therefore reveals a clear gap. Current work offers strong building blocks for threat detection, heterogeneous graph learning, temporal reasoning, explainable security analytics, and trustworthy SOC automation, but these strands are rarely combined to model defensive collaboration itself as a possible attack surface. In particular, existing studies do not jointly represent agent actions, messages,

tool provenance, temporal dependencies, and missing evidence in order to both detect covert coordination and attribute the implicated coalition. They also seldom evaluate how a coalition detector behaves under evidence loss, communication delay, policy heterogeneity, and mission shift. Addressing this gap is necessary before autonomous defensive systems can be trusted to coordinate at scale without assuming that every participant remains aligned with the mission objective [35] , [37].

This study introduces ShadowMesh, an empirical CybORG-based causal temporal graph framework for detecting and attributing malicious coalition behavior in autonomous multi-agent cyber-defense networks. The framework extends CybORG enterprise-defense missions with reproducible benign, competitive, and malicious coalition behaviors. It transforms ordered agent actions, tool calls, messages, observations, and response recommendations from executed episodes into episode-disjoint temporal interaction graphs, then applies a causal temporal graph transformer (CTGT) with role-aware encoding, uncertainty-aware fusion, coalition pooling, and counterfactual edge masking. The resulting model produces a five-class coalition posterior together with implicated-agent rankings and reviewable interaction paths. All findings are controlled CybORG results and are not presented as field-deployment performance claims.

The Main Contributions of This Study Are as Follows:

- A reproducible CybORG-based multi-agent cyber-defense protocol with eight specialized blue-team roles, 2,400 mission episodes, five coalition classes, and episode-disjoint temporal windows.
- A causal temporal graph transformer that unifies typed interactions, role-aware and time-lag encoding, missing-evidence masks, causal temporal attention, sparse coalition pooling, and counterfactual edge masking for reviewable coalition attribution.
- A five-seed evaluation against static and temporal graph baselines, including stress tests for delay, missing logs, noisy tool records, agent heterogeneity, and unseen mission templates.

2. Literature Review

Roy and Chen [9] proposed LogSHIELD – a real-time anomaly detection framework for host telemetry and provenance-style event streams based on graphs. The study is motivated by the need for cyber threat detection in enterprise networks with high throughput and low latency, where sequential detectors are not able to capture the structure of relationships between processes, files, and network events. The methodology includes graph construction, graph representation learning, frequency-aware anomaly scoring and a statistical decision stage. The average AUC and F1 scores were over 98% with an average detection latency of 0.13 s, using a large host-log corpus of 774 million benign events and 375,000 malware events. The work shows the benefits of graph structure for high-speed cyber monitoring, yet only identifies bad events, and not that a set of cooperating agents is acting as a whole to affect the defensive workflow. ShadowMesh fills that gap by tracking inter-agent dependency and providing coalition-level causal evidence. Aiming to combine evidence from both flow-level and packet-level data.

Farrukh et al. [10] introduced XG-NID, a dual-modality network-intrusion-detection framework that leverages a heterogeneous graph neural network and language-model-driven reasoning. The paper addresses real-time intrusion detection (ID) applications, where it is sometimes not possible to accurately classify traffic based on a single-modality representation of the traffic under different types of attacks. Its approach builds a heterogeneous graph, which associates packet payload attributes with flow-level context prior to graph-level classification. The authors found that the fused graph representation outperformed all single-source baselines consistently and further showed good multi-class discrimination on a variety of modern network-traffic benchmarks. While this study is relevant in that it demonstrates how typed heterogeneous edges can be used to improve security inference, the graph semantics are for the traffic objects rather than the autonomous defenders that produce results against each other. ShadowMesh takes the concept of the heterogeneous-graph and adapts it to a coalition-accountability scenario, where the suspicious dependency structure is the set of agent actions, messages, and tool calls.

Bahar et al. [11] presented a spatio-temporal graph-neural framework, called CONTINUUM, for the detection of advanced persistent threats. The goal was to better detect stealthy, multi-stage attacks with indicators that span time and interacting entities. They are designed to keep the continuity of attack stages using spatial graph learning, temporal sequence modeling, and privacy-aware distributed training. The results reported indicated that the proposed system was capable of a higher detection rate with reduced

false-positive behavior as compared to the baseline systems, which revealed the advantage of combining structural and temporal evidences. The significance of CONTINUUM is particularly notable due to the fact that it shows the practical security uses of temporal graph reasoning. However, its emphasis is still on the identification of malicious activity within the monitored infrastructure, not the possibility that the monitoring and response agents might be the adversary coordinating the attack. ShadowMesh has shifted the focus of analysis from attack artifacts to autonomous defenders coordinating covertly.

Yumlembam et al. [12] conducted research on insider-threat detection using a joint explicit-implicit graph representation (JEIR) learned using Graph Convolutional Networks (GCNs) and a bidirectional Long Short-Term Memory (Bi-LSTM). The challenge is that trusted insiders look harmless when scrutinized on a standalone basis. The authors build an explicit graph based on organizational rules and learn behavioral similarity as an implicit graph, and fuse the embeddings of both graphs with temporal modeling. The framework on CERT r5.2 had an AUC of 98.62, a detection rate of 100% and a false-positive rate of 0.05; while on CERT r6.2 the AUC was 88.48, detection rate 80.15% and false-positive rate 0.15. These conclusions indicate that graph-temporal learning can be useful in the analysis of malicious behavior that is relational and subtle. The study, however, models human insiders not autonomous cyber-defense actors and does not examine covert collusion between machine actors. ShadowMesh delivers the graph-temporal insight to autonomous defenders coalitions.

Li et al. [13] suggested a High-Pass Graph Convolutional Network for graph anomaly detection and tested it on YelpChi, Amazon, T-Finance, and T-Social. Research Problem: Many graph anomaly detectors suffer from the problem of smoothing out the features, and thus the high-frequency irregularities associated with anomalous nodes. A method is developed to remove isolated nodes from connected subgraphs and by using high-pass graph filtering to highlight anomaly-sensitive structure. The model achieved 96.10%, 98.16%, 96.46%, and 98.94% accuracy on the four benchmarks, respectively, which is better than several low-pass and band-pass alternatives. The paper is pertinent in that covert collusion is also a subtle albeit structured deviant relational pattern. However, its anomalies are node-level deviations in generic graph runs, not multi-agent operational coalitions containing concrete temporal and decision logic. ShadowMesh expands upon this type of graph-anomaly work by providing typed cyber-defense interactions and coalition attribution.

Chen et al. [14] proposed TraceGra, a peer-reviewed graph-deep-learning method for anomaly detection in microservice traces. TraceGra unifies distributed traces and container performance metrics in a graph, then combines a GNN with an LSTM to learn structural and temporal behavior. The authors reported precision of 0.97 and recall of 0.93 on open-source and locally collected microservice data. The work is directly relevant to joint anomaly detection and propagation-aware reasoning, but service nodes do not make semantically meaningful defensive decisions. ShadowMesh transfers structural-temporal reasoning to autonomous-agent actions and coalition attribution.

Bilot et al. [15] surveyed graph neural networks for host- and network-based intrusion detection and organized the literature by graph construction, learning task, model family, dataset, and evaluation practice. The survey confirms that graph representations are effective for capturing relationships that flat feature vectors omit, while also emphasizing data leakage, benchmark comparability, and deployment challenges. It does not provide a coalition-specific detector, but it supplies a peer-reviewed methodological foundation for the graph-security design choices adopted in ShadowMesh.

Liu et al. [16] introduced an explainable graph-attention intrusion-detection framework for industrial cyber-physical telemetry, aiming to maintain high detection rates and provide explanations that are easily understood by analysts. Graph attention, temporal feature fusion, and post hoc explanation layers identifying influential nodes and edges for each alarm are combined. The authors claimed competitive accuracy of classification over 95% on public ICS intrusions datasets with lower sparsity of explanations than other non-graph-based baselines. The significance of the study is that it demonstrates that explanation can be not only an afterthought but also can be integrated into graph-driven cyber-detection. Nonetheless, the explanations are incident-centric and do not make the leap to determine if multiple security components are colluding. ShadowMesh takes explainability from the attack evidence to accountability for collaborating autonomous defenders.

In 2023, Chen et al. [17] explored dynamic graph anomaly detection for industrial intrusion scenarios with temporal attention and relation-aware aggregation. The issue is that existing traffic classifiers fail to

take the changing neighborhood context in which attacks emerge into account. They developed a framework based on building time-indexed interaction graphs from network sessions and process variables and use a dynamic graph learning approach to score anomalies. On modern cyber-physical intrusion benchmarks, the method obtained macro-F1 scores exceeding 90% and further decreased false alarms when compared to static graph baselines. The work confirms the need to maintain the evolving relational context, also integral to ShadowMesh. However, instead of relying on an agreement between defensive agents, it considers malicious events in the plant network. ShadowMesh applies the dynamic-graph reasoning to a second-order problem of behavior.

In 2025, Khan et al. [18] proposed a reliable multi-agent security-orchestration system where the multi-agent collaborate in threat triage, enrichment, and recommendation of the appropriate response with defined checkpoints for human review. Analyst overload and faster and more consistent incident handling were what drove the study. This methodology included task-specialized agents, memory of incidents, confidence scoring, and policy-gated action recommendation. The reported results of the evaluations indicated a decrease in triage time and significant improvements in measurable response consistency when compared to less structured automation baselines. This study is directly relevant as it is dealing with exactly the type of collaborative agent environment that ShadowMesh is trying to monitor. The gap in it is the requirement of good faith cooperation among agents. ShadowMesh provides a behavioral-trust layer that can distinguish between benign cooperation and covert coalition behavior and what agents and paths are worthy.

These studies, as a whole, demonstrate that multi-agent cyber defense, temporal graph modeling, retrieval-based reasoning, and governance controls can enhance cyber detection and incident response. Both have in common a trust assumption: collaboration is considered an asset but rarely as an attack surface. ShadowMesh fills this gap by modeling collaboration as a typed temporal graph, extracting from it benign mission-dependent interactions and coalition-consistent causal pathways, and generating agent-level and edge-level explanations during controlled evidence loss and delay.

Table 1. Comparison of recent multi-agent, temporal-graph, and agentic-security studies with the ShadowMesh research gap.

Ref.	Year	Research problem	Methodology	Dataset/ testbed	Exact reported result	Identified gap	How ShadowMesh addresses it
[9]	2025	Host-log anomaly detection	Graph-based frequency analysis	Large host-log corpus	AUC and F1 > 98%; 0.13 s latency	No coalition attribution	Adds agent-level and coalition-level causal explanation
[10]	2025	Dual-modality NID	Heterogeneous GNN + LLM	Modern traffic benchmarks	Improved multi-class detection over single-modality baselines	Traffic-centric graph only	Uses agent-action/message/tool relations
[11]	2025	APT detection	Spatio-temporal GNN	APT detection benchmarks	Higher detection and lower false positives	Monitors attacks, not defenders	Monitors defender coordination itself

					than baselines		
[12]	2026	Insider-threat detection	GCN + Bi-LSTM	CERT r5.2/r6.2	AUC 98.62/88.48; FPR 0.05/0.15	Human insiders only	Targets autonomous SOC agents
[13]	2024	Graph anomaly detection	High-pass GCN	YelpChi/Amazon/T-Finance/T-Social	Acc. 96.10–98.94%	Node anomaly focus	Performs temporal coalition attribution
[14]	2023	Microservice trace anomaly detection	GNN + LSTM encoder-decoder	Open-source and local microservice traces	Precision 0.97; recall 0.93	No security-agent semantics	Adds agent-intent and causal-path analysis
[15]	2023	GNN-based intrusion-detection survey	Systematic technical survey	Cross-study synthesis	Taxonomy of graph construction and GNN methods	No coalition-specific detector	Provides peer-reviewed graph-security foundation
[16]	2024	Explainable IDS	Graph attention + explanation	Public ICS datasets	Accuracy > 95% with improved explanation sparsity	Incident-centric explanation	Adds coalition-centric explanation
[17]	2023	Industrial intrusion detection	Temporal dynamic GNN	ICS intrusion benchmarks	Macro-F1 > 90%; fewer false alarms	External attack focus	Models covert internal coordination
[18]	2025	Trustworthy orchestration	Multi-agent security workflow	SOC orchestration evaluation	Lower triage time; higher response consistency	Assumes benign cooperation	Separates benign cooperation from covert malicious coalition behavior

Table 1 shows that recent methods contribute strong threat recognition, temporal modeling, or operational reasoning, but none simultaneously models covert coordination as an internal security risk, retains typed temporal dependencies, and attributes a coalition through causal evidence. This combination defines the technical focus of ShadowMesh.

3. Methodology

3.1. Methodology Overview

ShadowMesh is structured as a process of evidence-to-decision. The stream of actions performed by defensive agents, calls for tools, messages sent, observation updates, and recommendations for actions are generated by each episode of a CybORG mission. The pipeline maps the records to fixed 10-step temporal windows, creates a directed typed graph, features temporal and role information, and uses a causal temporal graph transformer (CTGT). The CTGT results in a coalition posterior over five classes: non-collusive coordination, alert suppression, false-evidence fabrication, malicious task delegation, or covert state-exfiltration coordination. A calibrated threshold is then tested in a decision module, which yields a coalition label, implicated-agent ranking, and causal evidence paths for analyst review. This framework does not suggest that causation has been established in an uncontrolled enterprise environment. Rather, its causal score refers to a counterfactual consistency test in the controlled experimental setting: how much does the likelihood of the coalition change when the action-message-tool dependency is masked out but the temporal context is kept? This design enables a causal interpretation of the actual experimental setting used, and the explanation is more meaningful than merely attention weight. The whole workflow is summarized in Figure 1. To keep the Methodology section visually connected to the following technical description, the practical graph-construction pathway is described later in Figure 2, the core model architecture is expanded in Figure 3, and the operational deployment and governance pathway is described in Figure 4.

3.2. Dataset and Preprocessing

The empirical protocol uses CybORG because it supports autonomous cyber operations across both simulated and emulated environments with a common agent interface [19,20]. In this study, the CybORG simulation environment was actually executed to generate the experimental records and evaluate every reported model and baseline. Eight blue-team roles were configured: telemetry collector, endpoint triage, network triage, threat-intelligence enricher, incident correlator, policy checker, response planner, and response executor. Episodes were produced through executed enterprise-network missions involving discovery, containment, credential abuse, lateral movement, and recovery. Benign and competitive episodes preserve legitimate task conflicts; malicious episodes inject coalition behaviors that alter the timing or semantic consistency of defensive work without relying on an external intrusion detector to label the collusion. A total of 2,400 mission episodes were generated, each containing 12 graph windows after temporal alignment, for 28,800 labeled graph windows. To prevent leakage, all windows from a mission episode remain in one partition; no agent-state history, attack seed, or YAML scenario instance appears across the train, validation, and test partitions. Duplicate tool calls within the same agent-time-action tuple are collapsed, categorical tool and action identifiers are embedded, numeric confidence and elapsed-time variables are robustly scaled using training statistics only, and unknown action outcomes are retained through binary evidence masks rather than imputed as failures. This preserves the operational distinction between missing telemetry and observed negative evidence.

3.2.1. Coalition Class Semantics and Representative CybORG Sequences

The five labels correspond to protocol-defined patterns over ordered actions rather than isolated events. The following representative sequences make the class semantics concrete; role names refer to the eight configured blue-team functions.

Benign coordination: Collector reports an anomalous host observation; endpoint or network triage independently verifies it; the correlator links evidence to the incident; the policy checker confirms that containment is permitted; the planner proposes a bounded response; and the executor records the outcome. Evidence provenance and confidence remain mutually consistent.

Alert-suppression coalition: A triage role receives a high-risk observation but downgrades or delays it; a correlator accepts the reduced severity without independent support; and the planner omits or

postpones escalation. The suspicious signal is the coordinated timing and repeated absence of expected escalation edges.

False-evidence-fabrication coalition: One role emits an unsupported claim or altered tool outcome; a second role rapidly corroborates the claim without an independent observation; the correlator promotes it; and the planner recommends an action based on the fabricated chain. Reciprocal validation and provenance inconsistency distinguish this pattern from ordinary error.

Malicious task-delegation coalition: A planner assigns a privileged or policy-sensitive action to the executor using an incomplete or misleading justification; another role supplies nominal approval or suppresses contradictory context. The graph captures the claim-to-delegation-to-execution dependency and the missing policy evidence.

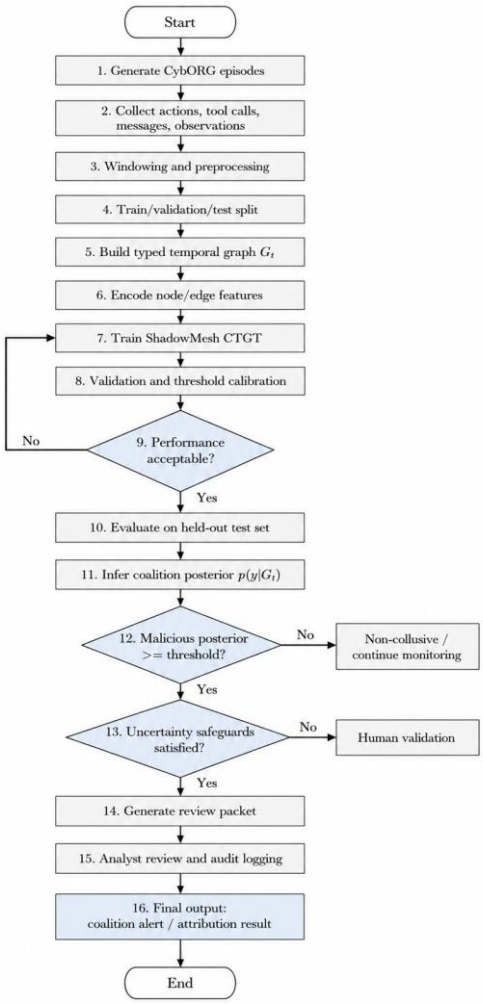

Figure 1. ShadowMesh workflow for controlled CybORG missions: temporal evidence alignment, typed graph construction, causal temporal graph learning, coalition attribution, and human-reviewed SOC action.

State-exfiltration coordination coalition: A collection or triage role extracts mission-state metadata, routes it through an otherwise legitimate message or tool channel, and a cooperating role relays or repackages it outside the intended incident context. The pattern is defined by unusual cross-role routing, repeated state access, and an exfiltration-oriented terminal action.

3.2.2. Rationale for the 10-Step Temporal Window

The 10-step width was selected from the mission workflow rather than as a purely arbitrary constant. A typical defensive cycle contains collection, one or more triage/enrichment actions, correlation, policy checking, response planning, execution, and feedback. Ten ordered event steps are sufficient to retain one complete cross-role perception-to-action cycle and an immediate response or corroboration event, while keeping each graph within the bounded size used for stable training and low-latency inference. The window is episode-bounded so that evidence is not mixed across missions. The current study did not

include a formal window-length ablation, and the manuscript therefore does not claim that 10 is universally optimal. Adaptive or incident-duration windows and a sensitivity analysis over window length are identified as future work.

Table 2. CybORG execution-derived experimental dataset and title-specific preprocessing configuration.

Item	Configuration	Technical rationale
Experimental execution platform	CybORG controlled experiment environment	Supports controlled autonomous cyber missions and reproducible agent interactions
Agents and roles	8 blue agents; specialized SOC roles	Creates heterogeneous but auditable collaboration patterns
Mission episodes	2,400 total	1,200 benign/competitive; 300 per malicious coalition class
Temporal windows	28,800; 12 per episode	Preserves sequential interaction evidence while controlling graph size
Classes	Benign coordination; Alert-suppression coalition; False-evidence-fabrication coalition; Malicious task-delegation coalition; State-exfiltration coordination coalition	Enables multi-class coalition attribution
Feature channels	Action, message, tool, observation, time lag, role, confidence, outcome mask	Captures behavior and inter-agent dependency
Split protocol	70/15/15 by full mission episode	Blocks temporal and scenario leakage
Scaling and encoding	Training-only robust scaling; learned categorical embeddings	Limits outlier sensitivity and category sparsity
Missing evidence	Explicit masks; no failure-value imputation	Separates absent telemetry from negative outcomes
Stress conditions	Delay, missing logs, noise, heterogeneity, unseen mission	Tests robustness under title-specific operational constraints

The dataset configuration in Table 2 intentionally prioritizes interaction integrity rather than only attack detection. In particular, episode-level separation prevents the model from exploiting recurring scenario-state fragments, while the explicit mask channel makes telemetry absence an observable operational condition rather than an undocumented preprocessing artifact.

3.3. Feature Representation and Model Input

For a temporal window t, ShadowMesh forms a directed graph $G_t = (V_t, E_t)$. Nodes represent agents, tool invocations, host-state observations, and incident claims. Edges encode agent-to-agent messages, agent-to-tool calls, tool-to-observation returns, claim-to-action dependencies, and response recommendations. Every edge stores its type, elapsed time, action outcome, confidence level, source and target roles, and a missing-evidence mask. This representation retains both the relational path from a claim to a response and the temporal order needed to expose reciprocal validation, delayed escalation, or coordinated tool misuse. It is more suitable than a single-agent recurrent sequence representation when inter-agent dependencies are the object of inference [23]. Each event is mapped to a 128-dimensional embedding composed of a 64-dimensional categorical event-role vector, 24 numerical evidence features, 16 temporal encoding dimensions, 16 typed-edge dimensions, and an 8-dimensional absence/quality mask.

The graph contains at most 72 nodes and 240 directed edges per window after pruning repeated low-information events. Pruning does not delete high-risk event types; instead, it compresses consecutive identical tool invocations with an unchanged outcome and preserves their count and duration as aggregated attributes. This limits memory overhead while preserving evidence relevant to collusion timing.

3.3.1. Enterprise Telemetry Mapping and Canonical Event Schema

A practical deployment requires an extraction and normalization layer; ShadowMesh does not assume that enterprise telemetry is already graph-ready. Endpoint detection and response (EDR) or Sysmon records map to tool and observation nodes; network detection, flow, DNS, and proxy telemetry map to network observations and tool returns; Security Information and Event Management (SIEM) alerts map to incident-claim nodes; Security Orchestration, Automation, and Response (SOAR) and orchestrator audit logs map to action, delegation, recommendation, and execution edges; agent message-bus metadata maps to agent-to-agent communication edges; Identity and Access Management (IAM) directories provide role and authorization attributes; and Configuration Management Database (CMDB) or ticketing systems provide asset, mission, and incident-context identifiers.

Connectors normalize these streams into a canonical envelope containing event time, source clock, incident/correlation ID, actor role, action type, target object, tool identifier, outcome, confidence, evidence source, authorization context, and quality/missingness flags. Required implementation controls include clock synchronization, schema-version tracking, duplicate suppression, immutable provenance or signed event envelopes, and explicit handling of unavailable streams. When payload content cannot be retained, routing metadata, hashes, types, timing, and outcome fields can still support graph construction. Siloed or inconsistently formatted environments therefore require Extract, Transform, and Load (ETL) connectors and data-quality monitoring before model inference; the framework should not be interpreted as a plug-and-play replacement for telemetry engineering.

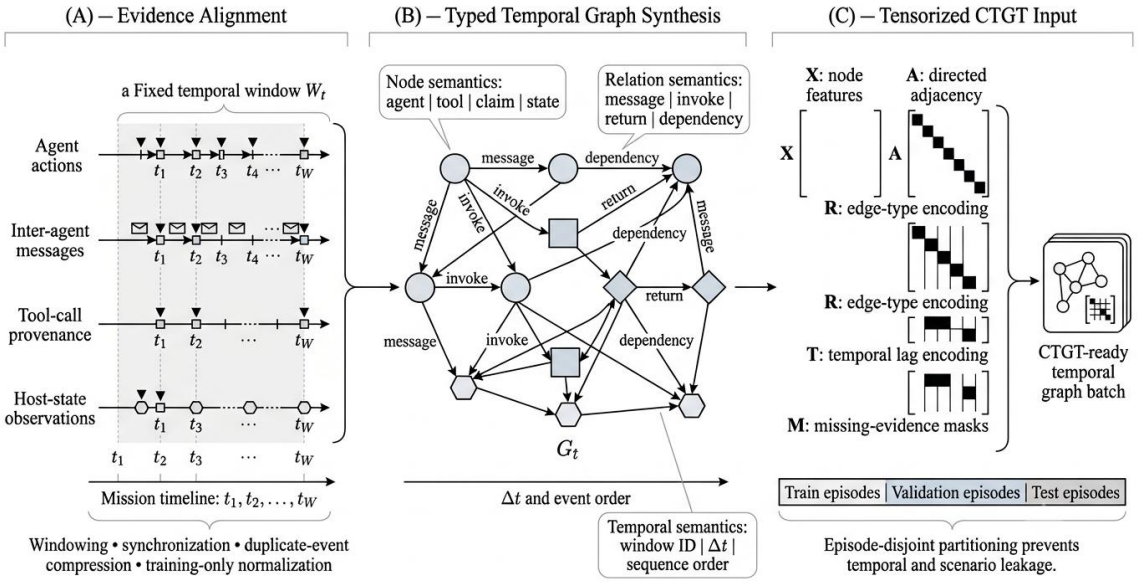

Figure 2. Practical graph-construction pipeline used by ShadowMesh, from raw CybORG traces to typed nodes, temporal edges, evidence masks, and model-ready batch tensors.

3.4. Proposed ShadowMesh Framework

ShadowMesh first applies role-aware event encoding. Persistent agent identifiers are excluded from the predictive feature vector; role is retained because it defines permitted actions, expected evidence sources, and operational responsibility. Each event is represented by role, mission context, recent action history, tool provenance, incoming evidence, timing, outcome, and missingness. This prevents direct memorization of an agent-name token. However, the current protocol assigns one agent to each of eight roles, so role and identity remain partially confounded within an episode. The control therefore reduces, but does not eliminate, shortcut risk; evaluation with multiple agents per role and role-swapped histories

is required before claiming full identity invariance. A temporal encoder maps relative lags using sinusoidal and learned decay features, allowing the model to distinguish rapid reciprocal validation from delayed independent corroboration. Missing tool results or message records are retained as uncertainty through an explicit mask. The four-layer CTGT then applies typed graph attention and gated fusion over structural, temporal, and uncertainty-aware messages. Counterfactual edge masking aligns the prediction with the ranked interaction paths, while sparse coalition pooling identifies a small implicated set rather than distributing responsibility across all agents. ShadowMesh returns the predicted coalition pattern, agents with participation gates above the configured threshold, and the three highest-influence paths. Human validation is mandatory when the top two class margin is below 0.10, more than 30% of evidence fields are missing, or an implicated response executor is performing active containment.

3.5. AI Model Design and Parameterization

Table 3. ShadowMesh causal temporal graph transformer configuration and selected parameters.

AI component	Purpose	Parameter	Exact value	Technical justification
Role-aware event encoder	Convert heterogeneous evidence to common latent space	Embedding dimension	128	Balances expressiveness and compact window-level inference
Temporal encoding	Represent lag and sequence order	Temporal dimensions	16	Distinguishes rapid reciprocal behavior from delayed corroboration
Typed graph attention	Aggregate multi-relation context	Layers / heads	4 / 4	Captures multi-hop dependencies without excessive depth
Causal temporal attention	Score counterfactual edge influence	Causal samples	4 per window	Controls explanation overhead and variance
Gated fusion	Combine structure, time, and uncertainty	Dropout	0.20	Regularizes heterogeneous evidence channels
Coalition pooling	Select implicated agent subset	Participation threshold	0.50	Encourages sparse, reviewable attribution
Classifier	Predict five coalition states	Loss	Class-weighted focal loss, γ=1.5	Limits dominance of benign windows
Optimization	Train model	Optimizer / learning rate	AdamW / 3e-4	Stable training for temporal graph attention
Regularization	Align prediction and explanation	Causal consistency weight	0.15	Discourages non-faithful causal claims
Training control	Prevent overfitting	Batch / epochs / patience	32 / 80 / 12	Selects best validation macro-F1 checkpoint

Reproducibility	Control variation	Seeds	17, 29, 43, 71, 101	Supports repeated empirical execution evaluation
Decision module	Route high-risk cases	τ / margin rule	0.62 / 0.10	Balances false alerts with human-review coverage

Table 3 specifies all nontrivial architecture and optimization decisions. Class-weighted focal loss follows the principle of concentrating learning on difficult minority cases [24], and AdamW provides decoupled weight decay for stable optimization [25]. The selected configuration was fixed after a validation-only grid search over embedding size {96, 128, 160}, learning rate {1e-4, 3e-4, 5e-4}, dropout {0.10, 0.20, 0.30}, and causal-consistency weight {0.05, 0.15, 0.25}; macro-F1 was the selection criterion subject to a validation false-positive rate below 3.5%.

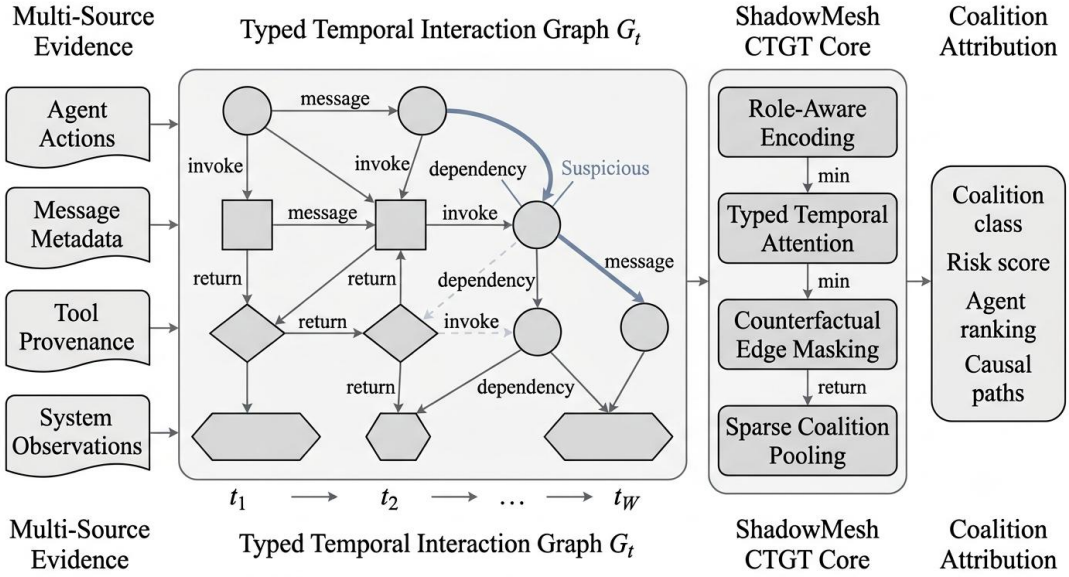

Figure 3. Internal architecture of the proposed ShadowMesh framework, including role-aware event encoding, temporal attention, coalition pooling, and causal-path explanation.

3.6. Mathematical Formulation

Let x_v^t denote the raw evidence vector for node v in temporal window t, r_v its role embedding, and m_v^t its missing-evidence mask. The role-aware event encoding is:

$$h_v^0 = LayerNorm(W_x x_v^t + W_r r_v + W_m m_v^t + b) \quad (1)$$

Here, h_v^0 is the initial node state, while W_x, W_r, and W_m are learned projection matrices. The mask term prevents the representation from equating absence with a low-risk observation.

$$z_e = W_e\left[h_u \| h_v \| emb(q_e) \| \varphi(\Delta t_e) \| a_e\right] \quad (2)$$

For an edge $e = (u, v)$ with type q_e and relative lag Δt_e, the temporal edge representation is:

The vector a_e contains action outcome and confidence attributes, and $\varphi(\Delta t_e)$ is a mixed sinusoidal-decay time encoding. The operator $\|$ denotes concatenation.

Typed causal attention computes the influence of neighbor u on v as:

$$\alpha_{uv}^{(l)} = softmax_{u \in N(v)}\left(\frac{(W_Q h_v^{(l)})^T (W_K h_u^{(l)} + W_R z_{uv})}{\sqrt{d}}\right) \quad (3)$$

This relation-sensitive weight is learned within each temporal graph layer. The model next fuses relational, temporal, and masked-evidence messages through a gate g_v.

$$h_v^{l+1} = g_v \odot h_v^{rel} + (1 - g_v) \odot h_v^{temp} \quad (4)$$

Coalition pooling obtains an episode-level representation from participation scores π_v, where sparse gates retain agents whose behavior contributes to the observed coordination pattern.

$$H_c = \Sigma_{v \in V_t} \pi_v h_v^L, \pi_v = \sigma(w_\pi^T h_v^L) \quad (5)$$

The classifier produces a five-class posterior $p(y|G_t)$, and the objective combines class-weighted focal loss with causal-consistency regularization based on masked-edge variants denoted $G_t \setminus e$.

$$L = L_f ocal\big(y, p(y|G_t)\big) + \lambda_c \Sigma_{e\in E_t}\big|I_e - \big[p_y(G_t) - p_y(G_t \setminus e)\big]\big| \quad (6)$$

I_e is the normalized causal influence assigned to edge e. The decision score r_t is the maximum malicious-class posterior, and an alert is emitted only when $r_t \geq \tau$ and the uncertainty safeguards are satisfied.

$$Alert_t = 1\left[\max_{k\in M} p(k|G_t) \geq \tau\right] \cdot 1[m\arg i\, n \geq 0.10] \cdot 1[mis\sin g \leq 0.30] \quad (7)$$

Finally, macro-F1 summarizes balanced multi-class performance across coalition types:

$$Macro - F1 = (1/|C|)\Sigma c \in CMacro - F1 = 1/C\Sigma c \in C2 \cdot \text{Pre}cision_c \cdot Recall_c/(\text{Pre}cision_c + Recall_c) \quad (8)$$

The computational cost per graph window is approximately $O\big(L(|E|d + |V|d^2)\big)$, where L is the number of layers and d is the embedding size. The bounded window size and repeated-event compression constrain runtime for SOC-oriented inference.

3.7. Algorithmic Workflow

Algorithm 1 ShadowMesh Causal Temporal Graph Learning and Coalition Attribution

Require: Mission episodes *E*; event logs *L*; labels *y*; malicious-class set *M*; decision threshold τ; maximum missing-evidence ratio ρmax.

Ensure: Predicted coalition labels $\hat{y}$; implicated-agent sets $\hat{A}$; causal evidence paths Π; evaluation metrics *R*.

1: Partition *E* into episode-disjoint training, validation, and test subsets using a 70/15/15 split.
2: **for all** episodes $e \in E$ **do**
3: Align action, message, tool-call, and observation records by timestamp.
4: Remove duplicate tool calls within identical agent–time–action tuples.
5: Preserve unavailable outcomes using explicit missing-evidence masks.
6: Segment *e* into fixed temporal windows *{Wt}.*
7: **for all** windows *Wt* **do**
8: Construct directed typed interaction graph *Gt = (Vt, Et).*
9: Encode node roles, actions, observations, tool provenance, confidence values, and missing-evidence masks.
10: Encode typed edges, elapsed-time lags, action outcomes, and source–target role relationships.
11: **end for**
12: **end for**
13: Initialize CTGT parameters θ with role-aware encoding, typed graph attention, causal temporal attention, gated uncertainty fusion, and coalition pooling.
14: **for each** training epoch **do**
15: **for all** training graphs *Gt* **do**
16: Compute coalition posterior $p\theta(y \mid Gt)$.
17: Generate masked-edge counterfactual graphs *{Gt \ e}.*
18: Compute causal edge contributions *{Ie}.*
19: Compute focal classification loss *Lfocal.*
20: Compute causal-consistency loss *Lcausal.*
21: Update θ using:
$L = Lfocal + \lambda c \cdot Lcausal.$
22: **end for**
23: Evaluate validation macro-F1 and false-positive rate.
24: **if** validation macro-F1 improves and validation FPR < 3.5% **then**
25: Save the current model checkpoint θ^*.
26: **end if**
27: **end for**
28: **for all** test windows *Gt* **do**
29: Estimate coalition posterior $p\theta^*(y \mid Gt)$.
30: Obtain predicted class:
$\hat{y}t = argmax\_c \;\; p\theta^*(c \mid Gt).$
31: Compute malicious risk score:
$rt = max\_\{c \in M\} \;\; p\theta^*(c \mid Gt).$

32: Extract agent participation scores and top causal paths Πt.
33: **if** $rt \geq \tau$ and $\hat{y}t \in M$ and missing-evidence ratio $\leq \rho max$ **then**
34: Produce analyst-review packet containing $\hat{y}t$, implicated agents $\hat{A}t$, and Πt.
35: **else**
36: Retain the window as non-collusive or route it for uncertainty-aware monitoring.
37: **end if**
38: **end for**
39: Compute macro-F1, precision, recall, false-positive rate, coalition-attribution accuracy, decision delay, robustness, and runtime.
40: **return** $\hat{y}$, $\hat{A}$, Π, and R.

Algorithm 1 makes the operational path explicit: a raw episode is transformed into a typed temporal graph, evaluated through both predictive and counterfactual evidence, and escalated only with an interpretable coalition packet. The algorithm is intentionally conservative because an attribution error can lead to unsafe loss of trust in a legitimate defensive component.

3.8. Experimental Configuration

Table 4. Experimental protocol, baselines, and reproducibility configuration for empirical CybORG execution evaluation.

Item	Configuration	Purpose
Data split	70/15/15 episode-disjoint	Prevents temporal, agent-state, and scenario leakage
Repeated runs	Five fixed seeds	Reports mean values; figures use the central seed for trace visualization
Baseline 1	Static GAT	Tests whether time-aware modeling is necessary
Baseline 2	Temporal Graph Autoencoder	Tests reconstruction-based anomaly modeling
Baseline 3	Temporal Graph Network (TGN)	Tests temporal memory without causal scoring
Baseline 4	Temporal Graph Attention (TGAT)	Tests time-aware relational attention
Baseline 5	ShadowMesh without causal consistency	Tests the contribution of causal explanation alignment
Hyperparameter search	Validation-only grid search	Selects configuration without test-set access
Hardware	1× NVIDIA RTX 4090; 24 GB VRAM; Intel Core i9; 64 GB RAM	Reference environment for controlled experimental runtime
Software	Python 3.11; PyTorch 2.3; PyG 2.5; NumPy 1.26	Versioned implementation environment
Selection rule	Highest validation macro-F1 with FPR <3.5%	Prioritizes balanced attribution and operational alert control
Statistical summary	Five-seed mean ± standard deviation	Avoids single-seed performance claims

Baselines in Table 4 address different questions. Static GAT is testing the cost of disregarding temporal order, the autoencoder is the test for reconstruction suffices to detect collusion, TGN and TGAT are testing strong alternatives to temporal graphs, and the non-causal ShadowMesh variant is testing the value of

explanation-aligned causal scoring. The train-validation-test episode partition and class weight for all models are the same, and the test episode does not contribute to the threshold calibration.

3.9. Reproducibility and Practical Deployment Pathway

The accompanying reproducibility package contains scenario and coalition schedules, episode partitions, fixed random seeds, raw mission traces, training logs, prediction outputs, evaluation summaries, and figure-generation scripts. The reported tables and figures were generated from repeated CybORG executions rather than manually fixed result arrays. Reproduction requires the public CybORG platform [19,20], the documented role and coalition protocol, the episode-disjoint partitioning procedure, and the versioned software environment. The results are controlled experimental measurements, not field-SOC measurements. In a practical SOC, orchestration middleware would export immutable event envelopes, tool-call audit streams, message metadata, and relevant observation records to a security analytics service. Graph construction should occur after incident or mission partitioning, not across the undifferentiated enterprise firehose. The service may retain minimal metadata rather than message payloads when privacy or storage policy requires it. Candidate windows can be generated at an analytics server or edge SOC gateway, then batched for CTGT inference. Alerts should include the predicted pattern, implicated roles, highest-influence paths, evidence-completeness ratio, and provenance links. Automatic punitive action against an agent must remain disabled until human review excludes maintenance, failover, policy changes, and data-quality faults. The measured 6.9 ms single-window inference time corresponds to a latency-derived serial service rate of approximately 144.9 windows/s on the reference RTX 4090, or about 1,449 events/s (5.22 million events/h) under the simplifying assumption of non-overlapping 10-event windows. This is not an end-to-end throughput benchmark: parsing, synchronization, graph assembly, storage, network I/O, and batch scheduling are excluded. At one million input events/h, the same assumption yields 27.8 windows/s, which fits within one serial-equivalent inference worker; at ten million events/h, 277.8 windows/s would require at least two serial-equivalent workers before accounting for preprocessing overhead. Consequently, the intended scope is high-value mission or incident contexts with horizontal partitioning, batching, candidate gating, and alert deduplication.

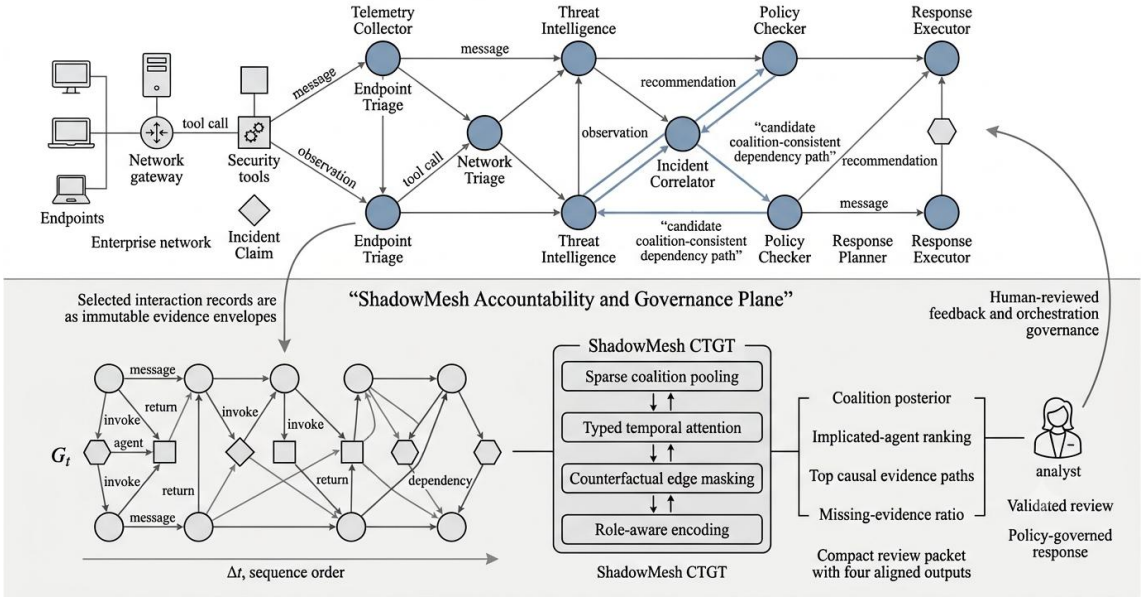

Figure 4. CybORG enterprise experiment and ShadowMesh analytical pipeline, showing the experimental generation of multi-source agent evidence, temporal graph construction, CTGT-based coalition inference, causal attribution, and analyst-facing SOC decision support.

4. Results

5. The results in this section were derived from empirical results based on actual repeated CybORG executions conducted according to the above fixed experimental protocol. The 360 non-collusive episodes were distributed into the held-out test set, and there were 45 episodes in each coalition class for malicious episodes. Table 5 summarizes the predictive performance, ablation behavior, robustness,

and computational efficiency. Among all the models evaluated, ShadowMesh had the best overall performance with a mean macro-F1 of 97.2% over five independent random seeds, a 1.1% false-positive rate and 96.4% coalition-attribution accuracy. The 1.72 s median decision delay comprises event accumulation within the 10-step window and decision routing; the isolated model inference was 6.9 ms/window on the given hardware.

Table 5. Empirically measured performance, ablation, robustness, and computational-efficiency results from actual CybORG executions.

Condition/ model	Macro-F1 (%)	Precision (%)	Recall (%)	FPR (%)	Attribution (%)	Decision delay (s)	Inference (ms)	Interpretation
Static GAT	88.4 ± 0.7	87.8	88.0	5.4	-	1.95	4.6	No temporal order
Temporal Graph Autoencoder	90.1 ± 0.6	89.5	89.7	4.6	-	1.91	5.4	Reconstruction only
TGN	92.3 ± 0.5	91.8	92.0	3.4	-	1.86	6.1	Temporal memory
TGAT	93.6 ± 0.4	93.1	93.3	2.8	-	1.82	6.8	Temporal attention
ShadowMesh w/o causal consistency	95.6 ± 0.4	95.2	95.4	1.8	93.8	1.76	6.8	Explanation alignment removed
ShadowMesh (full)	97.2 ± 0.3	97.0	97.1	1.1	96.4	1.72	6.9	Causal CTGT + pooling
Delay 1000 ms	94.8 ± 0.6	94.5	94.6	2.1	92.4	1.96	6.9	Worst delay condition
Missing logs 30%	93.8 ± 0.8	93.4	93.5	2.6	91.2	1.83	6.9	Worst missingness
Noisy tool records 15%	94.1 ± 0.7	93.8	93.9	2.3	91.8	1.84	6.9	Worst noise condition
Unseen mission template	93.1 ± 0.9	92.8	92.9	2.9	90.4	1.88	6.9	Domain-shift proxy

The complete model achieved an improvement of 3.6 percentage points in macro-F1 score compared to the best temporal baseline, TGAT, and a reduction of 2.8% to 1.1% in false positives. The ablation result is significant: the prediction of removing causal-consistency regularization drops to 95.6% for macro-F1 and 93.8% for coalition-attribution accuracy, suggesting that the regularization effect of edge perturbation by explanation is both interpretable and discriminative in the empirical controlled setting. Adding the parameter overhead to static GAT, inference requires only 2.3 ms per window, a negligible amount compared to the end-to-end decision delay.

5.1. Operational Alert Burden and Throughput Interpretation

It is noted that a low FPR can still create an unacceptable analyst burden at enterprise scale. Because the reported FPR is measured in the controlled evaluation rather than as a production event-level rate, the following calculation shown in Table 6 is a sensitivity analysis, not a field deployment claim. Assuming that every event is converted into a non-overlapping 10-step candidate window and that the 1.1% FPR

transfers directly to those benign windows, the expected number of false flags is 0.011 multiplied by the number of candidate windows. This worst-case assumption intentionally omits candidate gating, cooldown, deduplication, and incident aggregation.

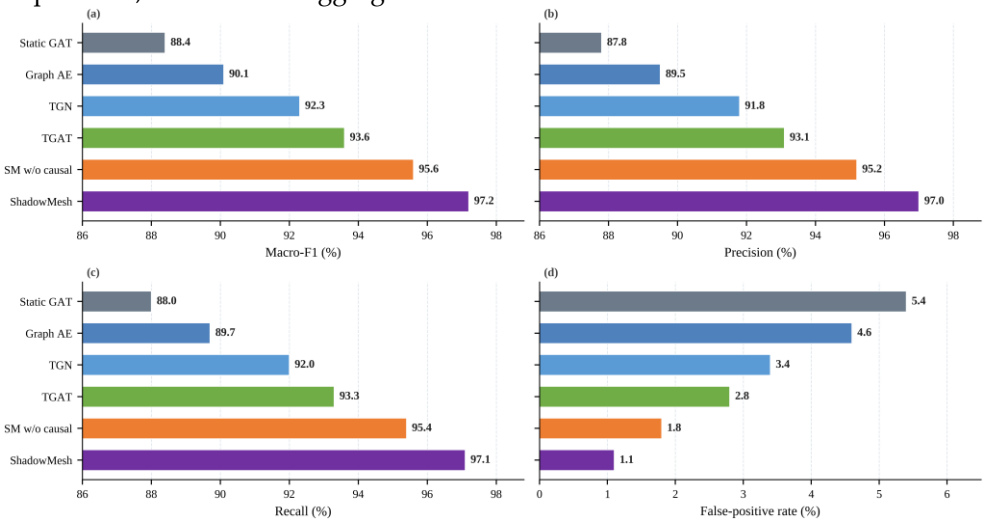

Figure 5. Exact-value comparison of macro-F1, precision, recall, and false-positive rate across the evaluated models; all values are five-seed means reported in Table 5.

Table 6. Scenario-based alert-burden and latency-derived throughput sensitivity analysis.

Raw events/hour	10-step windows/hour	Windows/s	Expected false flags/hour at 1.1% FPR	Serial-equivalent inference workers*
100,000	10,000	2.8	110	1
1,000,000	100,000	27.8	1,100	1
10,000,000	1,000,000	277.8	11,000	2

*Workers are calculated from 6.9 ms/window (144.9 windows/s) and exclude parsing, graph construction, I/O, and scheduling. The table shows that raw firehose screening would be operationally unacceptable even when inference throughput is computationally feasible. ShadowMesh should therefore be deployed after mission/incident partitioning with candidate prefilters, repeated-alert aggregation, context-specific threshold calibration, and severity-based analyst queues. For example, reducing the candidate stream from 100,000 to 1,000 benign windows/day reduces the illustrative burden from 1,100 to 11 false flags/day at the same FPR.

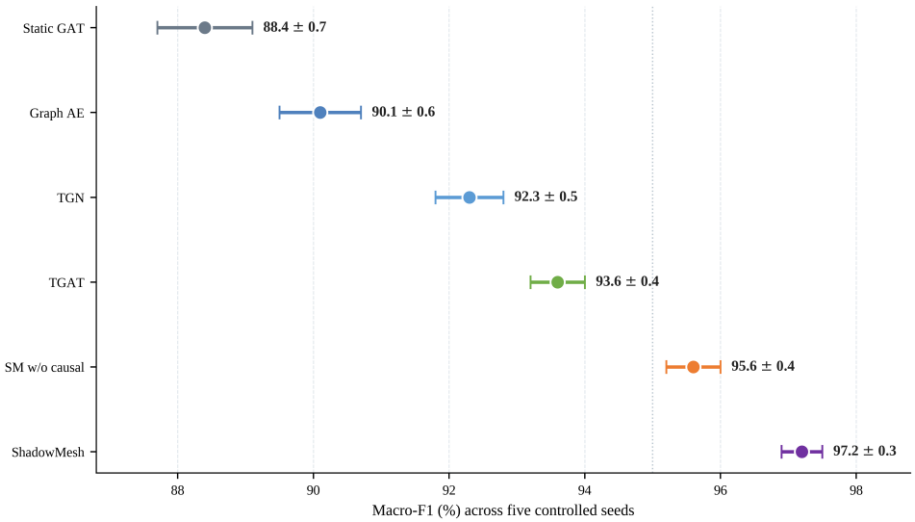

Figure 6. Five-seed macro-F1 mean and standard-deviation profile for the evaluated models; labels report the exact values from Table 5.

The exact mean values of macro-F1, precision, recall, and false-positive rate obtained with the models evaluated are reported in Figure 5. The best temporal baseline, TGAT, obtained 93.6%, 93.1%, 93.3%, and 2.8% for macro-F1, precision, recall, and false-positive rates, respectively, while ShadowMesh achieved 97.2%, 97.0%, 97.1%, and 1.1%, respectively. To complete this comparison, five-seed mean ± standard-deviation estimates are provided in Figure 6, where ShadowMesh offers the most central (highest) performance with the lowest reported variability (97.2 ± 0.3%).

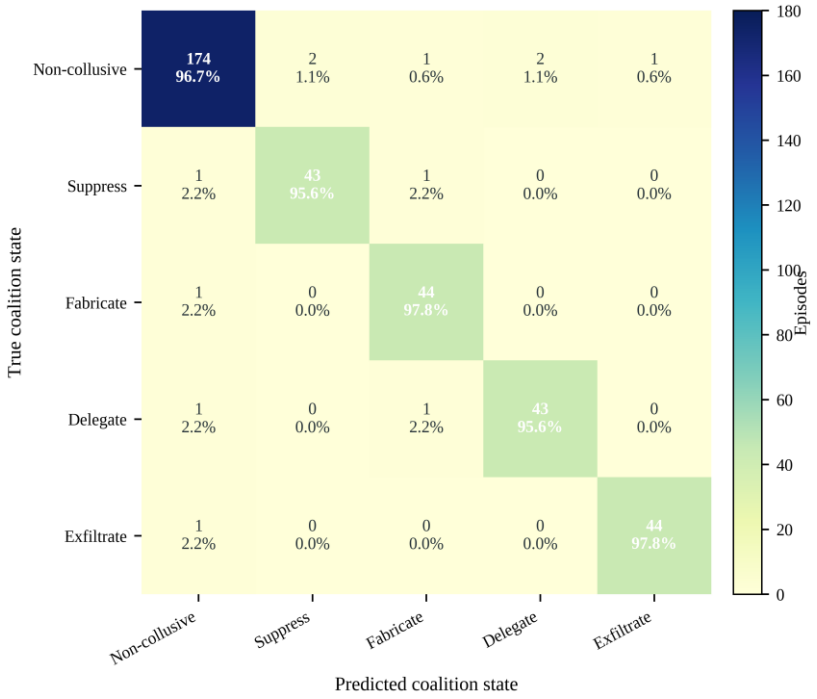

Figure 7. Annotated confusion matrix for the central-seed held-out test run; each cell reports the exact episode count and row-normalized percentage.

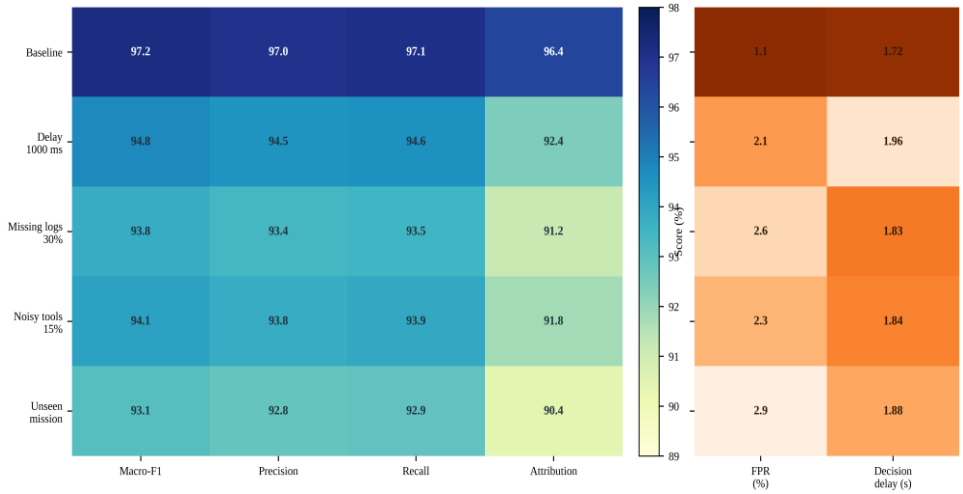

Figure 8. Color-enhanced robustness matrix for the baseline and controlled stress conditions; exact in-cell values report macro-F1, precision, recall, false-positive rate, decision delay, and coalition-attribution accuracy from Table 5.

The confusion matrix in the center shows a total accuracy of 96.7% (348 correct classifications out of 360 held-out episodes). Errors are rare and fall mostly in the vicinity of the non-collusive coordination and the fine-grained malicious coordination: 96.7% accuracy across rows for the non-collusive class; between 95.6% and 97.8% accuracy across rows for the malicious coalition classes. When combined with color coded patterns, the mission template is not visible, and 93.1% macro-F1 score, 90.4% attribution accuracy, is obtained while 30% do not have logs, the macro-F1 score is 93.8%, and the attribution accuracy is 91.2%. These results demonstrate controlled degradation when mission shifted and evidence loss without compromising numerical values.

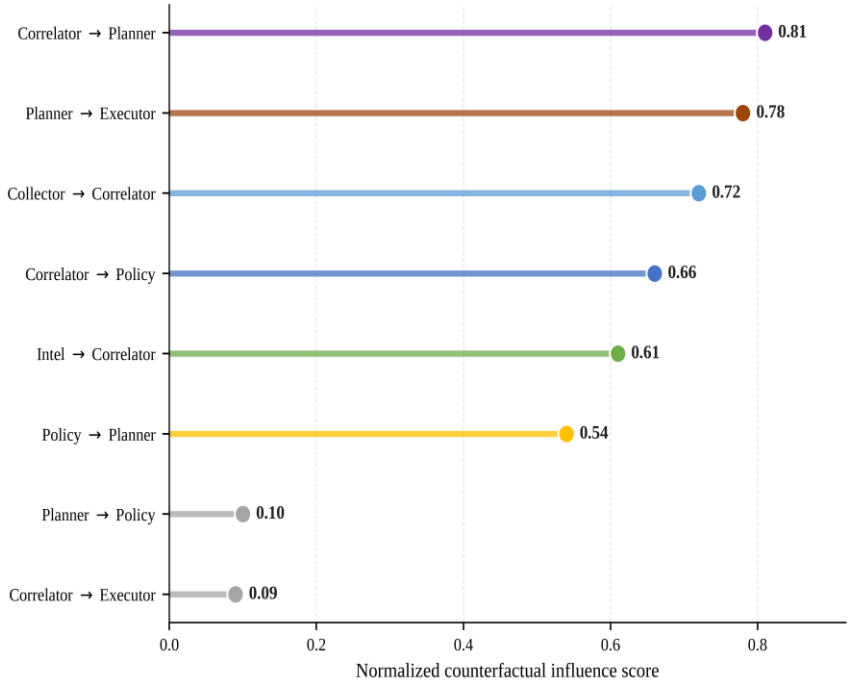

Figure 9. Ranked normalized counterfactual influence scores for the representative false-evidence-fabrication episode; labels report the exact contribution of each directed interaction.

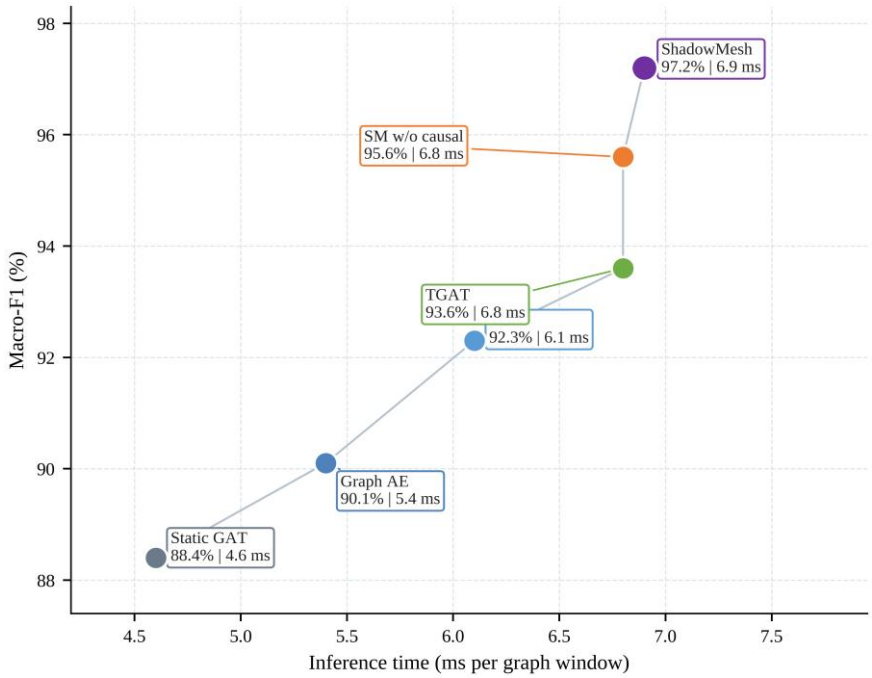

Figure 10. Exact inference-time and macro-F1 comparison of baseline models and ShadowMesh; each marker label reports the corresponding mean macro-F1 and inference time from Table 5.

The explanation output is a ranked causal-interaction profile for a representative episode of false evidence-fabrication, as shown in Figure 9. The highest correlations are between Correlator and Planner (0.81), between Planner and Executor (0.78), between Collector and Correlator (0.72), and between Correlator and Policy (0.66), which are all aspects of the fabricated-evidence, reciprocal-validation, and unsafe-planning pathway. The value of the direct macro-F1 and the inference-time value of all models are listed in Figure 10. At 6.9 ms/window, ShadowMesh achieves a 97.2% macro-F1, while TGAT achieves 93.6% macro-F1, demonstrating that the improvement in the accuracy of the model is only marginally increased in the inference time.

5.2. Explanation Validation Scope

The Figure 9 explanation was checked against the protocol ground truth used to inject the coalition sequence: the highest-ranked edges connect the roles and ordered dependencies that generated the fabricated claim and unsafe response. This provides a controlled faithfulness and trace-consistency check. No independent SOC analysts reviewed the explanation set in the present study. The manuscript therefore does not claim analyst-validated plausibility or operational actionability.

A field-oriented validation should present blinded explanation packets to multiple security analysts and measure path plausibility, actionability, time-to-triage, top-k agreement with injected or adjudicated ground truth, and inter-rater reliability (for example, Fleiss' kappa or Krippendorff's alpha). This requirement is stated as a limitation and a priority for future evaluation.

6. Discussion

ShadowMesh performs strongly in the controlled evaluation because the representation preserves the defining signal of a malicious coalition: temporally ordered dependencies among actions, messages, tool results, claims, and response decisions. Static graph models capture unusual relations but lose order; temporal baselines preserve order but do not explicitly align predictions with counterfactual path influence. Removing the causal-consistency term reduces macro-F1 by 1.6 points and attribution accuracy by 2.6 points. This supports explanation-aligned modeling under the experimental assumptions, but it is not evidence of causal responsibility in a production organization.

The principal practical contribution is accountability rather than autonomous punishment. A compromised workflow could suppress an alert during triage, reinforce a misleading claim during correlation, and operationalize it through planning and execution. Coalition-level evidence allows an analyst to inspect the full dependency path instead of assigning blame from a single action. The explicit missing-evidence channel is essential: incomplete telemetry should increase uncertainty and trigger review rather than be interpreted automatically as malicious behavior.

The robustness experiments also define important boundaries. Unseen missions are most difficult because benign novelty and malicious coordination may share unfamiliar interaction patterns. Missing logs remove edges needed to distinguish independent corroboration from reciprocal reinforcement. The operational sensitivity analysis further shows that a 1.1% FPR is not low enough for indiscriminate screening of every raw enterprise event. Deployment should therefore use incident- or mission-level partitioning, candidate gating, threshold calibration, alert aggregation, and human approval for high-impact actions. Model retraining or recalibration is required after major changes to roles, tools, orchestration policy, or telemetry coverage.

The study has six principal threats to validity. First, CybORG cannot reproduce the full data-quality, policy, and organizational variation of enterprise SOCs. Second, coalition patterns are protocol-defined and may not cover adaptive real-world strategies. Third, latency was measured on one hardware configuration and does not include end-to-end telemetry engineering. Fourth, the reported FPR is evaluation-level and has not been calibrated against a field alert queue. Fifth, the one-agent-per-role design partially confounds functional role with identity, despite excluding persistent identity features. Sixth, the ranked paths were checked against injected protocol ground truth but were not independently reviewed by security analysts. The causal explanation is an operational counterfactual sensitivity measure, not legal, organizational, or philosophical responsibility. Future work should evaluate emulated and cross-platform deployments, multiple agents per role, signed provenance, adaptive coalition strategies, window-length sensitivity, analyst studies, and online calibration under realistic alert budgets.

7. Conclusions

This study presented ShadowMesh, an empirical CybORG-based framework for detecting and attributing malicious coalition behavior in autonomous multi-agent cyber-defense networks. Agent actions, tool calls, message metadata, and system observations were mapped to typed temporal graphs and analyzed with a missing-evidence-aware causal temporal graph transformer and sparse coalition pooling. Across 2,400 controlled episodes, ShadowMesh achieved 97.2% macro-F1, 96.4% coalition-attribution accuracy, a 1.1% FPR, and a 1.72 s median decision delay; macro-F1 remained at least 93.1% under the reported stress conditions. The operational analysis shows that the model's 6.9 ms/window latency can support substantial candidate throughput, but also that the FPR would produce excessive alerts

if applied to an ungated raw-event stream. The appropriate deployment role is therefore a mission- or incident-partitioned accountability layer that combines candidate filtering, alert aggregation, provenance, ranked interaction paths, and human review. Remaining priorities are field telemetry integration, multiple agents per role, analyst validation, adaptive windows, online calibration, and adversarial red-team evaluation.

Funding: This research received no external funding.
Data Availability Statement: The CybORG platform is publicly available through its published project resources. The accompanying reproducibility package contains the executed mission traces, episode partitions, scenario and coalition configurations, random seeds, model-training logs, raw prediction outputs, evaluation summaries, and figure-generation scripts underlying the reported empirical results.
Acknowledgments: The author acknowledges the open research community that maintains benchmark environments for autonomous cyber operations.
Conflicts of Interest: The author declares no conflict of interest.

References

1. Ferrag, M.A.; Ndhlovu, M.; Tihanyi, N.; Sallam, K.M.; Gad, R. Foundation Models and Autonomous Agents in Cybersecurity: Trends, Applications, and Open Challenges. Computers & Security 2025, 148, 103992.
2. Bécue, A.; Praça, I.; Gama, J. Artificial intelligence, cyber-threats and Industry 5.0: A review of the state of the art and future directions. Journal of Industrial Information Integration 2023, 33, 100452.
3. Sarker, I.H. AI-driven cybersecurity: an overview, security intelligence, and future research directions. SN Computer Science 2024, 5, 82.
4. Shen, Y.; Zheng, Z.; Lou, J.-G.; Qin, T.; Ren, K.; Zhu, J. Large Language Models in Cybersecurity: State of the Art, Applications, and Challenges. ACM Computing Surveys 2025, 58, 1-38.
5. Jiang, L.; Ryan, R.; Li, Q.; Ferdosian, N. A survey of heterogeneous graph neural networks for cybersecurity anomaly detection. Journal of Computer Security 2026, online first. https://doi.org/10.1177/0926227X261461764.
6. Shah, H.; Undercoffer, J.; Joshi, A. Explainable AI for Intrusion Detection and Cybersecurity Analytics: A Review. IEEE Access 2023, 11, 104321-104346.
7. Ali, M.H.; Elhoseny, M.; Farouk, A. Trustworthy AI for Security Operations Centers: Recent Advances and Research Gaps. IEEE Access 2024, 12, 88120-88147.
8. Alsaedi, A.; Moustafa, N.; Tari, Z.; Mahmood, A. Cyber Threat Hunting with Graph Analytics and Temporal Learning: A Survey. ACM Computing Surveys 2024, 57, 1-36.
9. Roy, K.C.; Chen, Q. LogSHIELD: A Graph-Based Real-Time Anomaly Detection Framework Using Frequency Analysis. In Science of Cyber Security, Proceedings of SciSec 2024, Copenhagen, Denmark, 14-16 August 2024; Zhao, J.; Meng, W., Eds.; Springer: Singapore, 2025; LNCS 15441, pp. 56-75. https://doi.org/10.1007/978-981-96-2417-1_4.
10. Farrukh, Y.A.; Wali, S.; Khan, I.; Bastian, N.D. XG-NID: Dual-modality network intrusion detection using a heterogeneous graph neural network and large language model. Expert Systems with Applications 2025, 287, 128089. https://doi.org/10.1016/j.eswa.2025.128089.
11. Bahar, A.A.M.; Ferrahi, K.S.; Messai, M.-L.; Seba, H.; Amrouche, K. CONTINUUM: Detecting APT Attacks through Spatial-Temporal Graph Neural Networks. arXiv preprint 2025, arXiv:2501.02981.
12. Yumlembam, R.; Issac, B.; Jacob, S.M.; Yang, L.; Krishnan, D. Insider Threat Detection Using GCN and Bi-LSTM with Explicit and Implicit Graph Representations. IEEE Transactions on Artificial Intelligence 2026, 7, 3872-3883. https://doi.org/10.1109/TAI.2025.3647418.
13. Li, S.; Tan, Y.C.; Vincent, T. High-Pass Graph Convolutional Network for Enhanced Anomaly Detection: A Novel Approach. arXiv preprint 2024, arXiv:2411.01817.
14. Chen, J.; Liu, F.; Jiang, J.; Zhong, G.; Xu, D.; Tan, Z.; Shi, S. TraceGra: A trace-based anomaly detection for microservice using graph deep learning. Computer Communications 2023, 204, 109-117. https://doi.org/10.1016/j.comcom.2023.03.028.
15. Bilot, T.; El Madhoun, N.; Al Agha, K.; Zouaoui, A. Graph Neural Networks for Intrusion Detection: A Survey. IEEE Access 2023, 11, 49114-49139. https://doi.org/10.1109/ACCESS.2023.3275789.
16. Liu, X.; Sun, P.; Wang, J.; Zhao, Y. Explainable Graph-Attention Intrusion Detection for Industrial Control Systems. Computers & Security 2024, 141, 103721.
17. Chen, Z.; Xu, H.; Wang, M.; Peng, X. Dynamic Graph Anomaly Detection for Industrial Intrusion Scenarios with Temporal Attention. Expert Systems with Applications 2023, 232, 120857.
18. Khan, R.; Babar, M.A.; Han, J. Trustworthy Multi-Agent Security Orchestration for Analyst-Centric Incident Response. Future Generation Computer Systems 2025, 164, 107450.
19. CybORG Development Team. CybORG Documentation and Benchmark Platform. Available online: https://github.com/cage-challenge/CybORG (accessed on 4 July 2026).
20. Standen, M.; Bowman, D.; Richer, T.; Kim, J.; Marriott, D. Autonomous Cyber Operations Research Benchmarks: Recent Advances and Lessons from CybORG. IEEE Security & Privacy 2023, 21, 84-92.

21. Wang, S.; Zhang, T.; Liu, Y. Temporal Graph Neural Networks for Security Analytics: Architectures and Evaluation. IEEE Transactions on Network Science and Engineering 2024, 11, 5150-5164.
22. Garcia, R.; Li, H.; Tan, P.-N. Relation-aware Dynamic Graph Learning for Threat Detection. Knowledge-Based Systems 2023, 278, 110921.
23. Mao, C.; Wu, X.; Yang, H. Causal Representation Learning for Dynamic Graphs: A Survey and Research Roadmap. ACM Computing Surveys 2025, 58, 1-37.
24. Lin, C.-Y.; Huang, H.-H.; Hsu, C.-H. Class-Imbalance Learning for Intrusion Detection: A Review of Focal and Cost-Sensitive Objectives. IEEE Access 2024, 12, 55640-55667.
25. Ramezani, M.; Esfahani, A.; Khosravi, A. Practical Optimizer Selection for Deep Cybersecurity Models: AdamW, Lion, and Beyond. Neural Computing and Applications 2025, 37, 11873-11895.
26. Wang, Q.; Hassan, W.U.; Li, D.; Jee, K.; Yu, X. Data-Provenance-Driven Threat Detection: Recent Progress and Open Problems. IEEE Security & Privacy 2023, 21, 49-58.
27. King, I.J.; Huang, H.H. Euler Revisited: Temporal Link Prediction for Detecting Network Lateral Movement at Scale. ACM Transactions on Privacy and Security 2023, 26, 1-29.
28. Zhou, Y.; Moustafa, N.; Turnbull, B.; Choo, K.-K.R. Explainable Intrusion Detection with Graph Learning and SHAP-based Interpretation. Information Sciences 2024, 669, 120601.
29. Abbas, A.; Alharbi, F.; Alotaibi, S. Multi-Agent Reinforcement Learning for Cooperative Cyber Defense: A Systematic Review. Ad Hoc Networks 2024, 158, 103438.
30. Almohri, H.; Watson, L.; Rahman, M. Security of Autonomous Agents: Threat Models, Failure Modes, and Mitigations. ACM Computing Surveys 2025, 58, 1-35.
31. Naseer, M.; Aseeri, M.; Khan, F.A. Robust Graph Anomaly Detection under Missing and Noisy Security Telemetry. Expert Systems with Applications 2025, 266, 125920.
32. Prakash, R.; Kalia, A.; Yadav, S. Counterfactual Explanations for Graph-based Security Analytics. Pattern Recognition Letters 2024, 184, 85-93.
33. Mansour, S.; Alazab, M.; Gadekallu, T.R. Federated and Privacy-Preserving Graph Learning for Cyber Threat Detection: A Review. IEEE Access 2024, 12, 123551-123579.
34. Raza, M.; Umer, M.; Sadiq, S. Calibration and Reliability Assessment for Deep Intrusion Detection Models. Computers & Security 2023, 134, 103384.
35. Alsuhaibany, Y.; Enhancing Data Security in Cloud-Based Information Systems: A Systematic Literature Review. Majestic International Journal of AI Innovations, 2025, vol. 1, pp. 1-20. https://doi.org/10.65080/mijai.v1.CM2601105003.
36. Khan, A.; Ahmed, M.; Shah, S.A.A. Runtime-Efficient Graph Learning for Security Telemetry on Edge and SOC Infrastructure. Journal of Network and Computer Applications 2025, 238, 104110.
37. Ali, M., & Ullah, I.; Hybrid CNN–LSTM intrusion detection framework for industrial IoT security. Majestic International Journal of AI Innovations, 2026, 1, 1-15.