A Cross-Modal Deep Learning Framework for Joint Semantic and Structural Threat Detection in Provenance Graphs[*]

Seong-Su Yoon, Dong-Hyuk Shin, Hui-Seok Yang,
and Ieck-Chae Euom[†]

Chonnam National University, Gwangju, Republic of Korea
{skymoonight, shindh2, ysa8275, iceuom}@jnu.ac.kr

Abstract

Industrial control systems face sophisticated Advanced persistent treats that evade conventional detection techniques. Current provenance-based solutions are also limited, adopting a single-modality approach that analyzes temporal or structural patterns independently, thus compromising detection effectiveness. This paper presents a novel cross-modal deep learning architecture that jointly learns temporal progression and structural interaction patterns from provenance data via cross-modal attention and adaptive fusion. Evaluated on industrial control system data, the proposed architecture achieves over 93% accuracy and 95% precision, while cutting false positive rates by 72% relative to state-of-the-art techniques. This substantial reduction in false alarms mitigates a key deployment barrier, enabling practical adoption of automated threat detection in production industrial control system environments.

Keywords: Industrial control systems, cybersecurity, provenance analysis, cross-modal learning, deep learning, advanced persistent threats

1 Introduction

The cybersecurity landscape for industrial control systems (ICS) has evolved fundamentally, with critical infrastructure now a prime target for sophisticated threat actors. The frequency of attacks on critical infrastructure surged by nearly 70% in 2024, with actors like “CyberArmyofRussia_Reborn” and malware like FrostyGoop demonstrating targeted operations designed to compromise the availability and integrity of critical services[1, 2, 3].

Advanced persistent threats (APTs) in ICS environments have distinct characteristics. Unlike conventional malware, ICS-targeted APTs operate via multi-stage campaigns to achieve operational manipulation. While average detection times have improved to hours, advanced actors still evade detection by using living-off-the-land (LoTL) tactics and mimicking routine operational behaviors[4, 5]. These attacks exploit the unique characteristics of industrial networks, including protocol diversity, real-time constraints, and legacy system integration.

Existing security approaches for ICS face fundamental limitations. Conventional signature-based systems fail against the zero-day exploits and novel variants common in modern APTs[6]. Statistical anomaly detection

[*] Proceedings of the 9th International Conference on Mobile Internet Security (MobiSec'25), Article No. 27, December 16-18, 2025, Sapporo, Japan. © The copyright of this paper remains with the author(s).

[†] Corresponding author

methods can identify deviations from normal behavior but often produce high false positive rates, struggling to differentiate malicious activity from legitimate operational variability. A persistent gap remains between threat sophistication and defensive capabilities, with only 56% of organizations having ICS specific incident response plans[4].

Provenance-based security analysis has emerged as a promising method, modeling system behavior as directed acyclic graphs (DAGs) that encode causal dependencies. However, a critical limitation in existing provenance methods is the modality gap. Causal path-based approaches (using Long Short-Term Memory (LSTM), Gated Recurrent Unit (GRU), Transformer) excel at capturing the logical progression of attack chains but miss the global structural context. Conversely, global graph-based techniques (using Graph Neural Network (GNNs)) model holistic structural patterns effectively but fail to capture the temporal and causal sequencing of multi-stage attacks.

To overcome these challenges, this paper proposes a unified deep learning framework that combines causal and structural modeling to eliminate the modality gap. Our cross-modal architecture concurrently processes provenance data as both sequences of causal paths and global structural representations. The approach integrates three key innovations: (1) a dual-branch architecture with Transformer and Graph Attention Network (GAT) branches to extract causal and structural patterns, respectively; (2) cross-modal attention to foster bidirectional information exchange between modalities; and (3) comprehensive security feature engineering incorporating 156 relevant indicators.

The dual-branch architecture prevents information loss by preserving both causal progression and global structural relationships. Through cross-modal attention, the model simultaneously leverages causal patterns during structural analysis and incorporates structural context during sequence processing. Advanced fusion mechanisms, including learnable gating networks, then dynamically balance information from both modalities based on attack characteristics.

This study makes four primary contributions:

- **Novel Cross-Modal Architecture:** We introduce the first deep learning framework to jointly learn causal path progression and global structural patterns of ICS attacks, enabling a unified analysis that single-modality methods cannot achieve.
- **Advanced Fusion Mechanisms:** We developed learnable gating and cross-attention mechanisms that adaptively balance causal and structural information. Ablation studies show a 4.1% performance gain attributable solely to this cross-modal fusion.
- **Comprehensive Security Feature Engineering:** Our framework systematically uses 156 domain-specific indicators—spanning process, file, network, and temporal patterns—to capture malicious behavior in provenance data.
- **Practical Deployment Validation:** With 93.96% accuracy and 95.31% precision on ICS datasets, our framework reduces false positives by 72%, directly alleviating alert fatigue and enabling feasible deployment in production environments.

2 Background

Provenance graphs are effective for differentiating malicious and benign applications in operational technology (OT) environments. However, existing approaches suffer from fundamental limitations by considering only either sequential causal relationships or structural connectivity patterns. This section demonstrates the critical importance of cross-modal analysis by examining concrete attack scenarios and their behavioral patterns, which provide insights into how sophisticated threats exploit the blind spots of single-modality approaches. Via detailed analysis of the Triton/TRISIS attack case[7,8], this study reveals how identical path sequences can mask malicious intent when structural context is ignored, and conversely, how structural patterns lose their significance without understanding the causal progression enabling them.

2.1 Challenges in Single-Modality Threat Analysis

2.1.1 Blind Spot 1: The Failure to See Structure in Sequences. Figure 1 demonstrates a challenge where two scenarios generate identical causal relationship sequences but have fundamentally different security implications. We examined the prevalent TRISIS/Triton attack that targeted a petrochemical facility, alongside a routine maintenance operation in the same OT environment [7, 8].

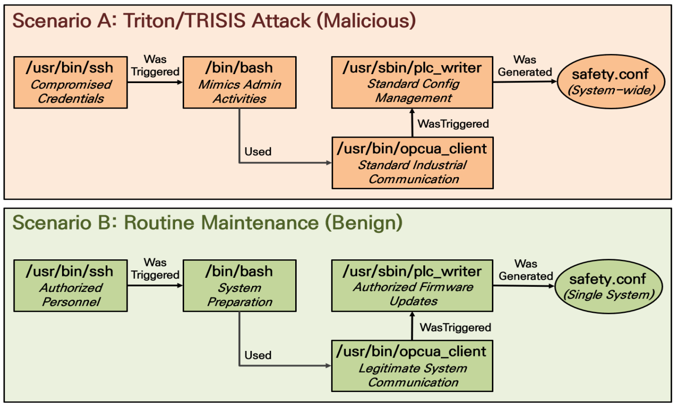

Figure 1: Limitation of Sequential Analysis - Identical Causal Relationship Sequences Between Triton/TRISIS Attack and Routine Maintenance Operations

The Triton attack (Scenario A) demonstrates a sophisticated utilization of legitimate industrial workflows. After establishing Secure Shell (SSH) access via compromised engineering credentials, the attacker executes bash commands that mimic normal system administration activities.

Subsequently, the attacker invokes opcua_client and plc_writer, tools that are commonly used in industrial environments for legitimate engineering and maintenance tasks. opcua_client denotes a client-side application implementing the Open Platform Communications Unified Architecture (OPC-UA) protocol, which is routinely used to query or update Programmable Logic Controllers (PLCs) and other field devices. plc_writer refers to a PLC maintenance/diagnostic utility used to write configuration or control values into PLC memory blocks. While these tools normally support benign operations (e.g., parameter tuning or firmware updates), the adversary abuses them to transmit unauthorized payloads and to overwrite safety logic with a malicious configuration file (e.g., safety.conf).

A benign maintenance operation (Scenario B) follows the same sequence: an authorized engineer gains SSH access, runs administrative commands, then uses opcua_client and plc_writer as part of a legitimate firmware or configuration update. Because the linear "sequence representation" (SSH → bash → OPC-UA client → PLC writer) is identical for both scenarios, sequence-only detectors that rely solely on ordered event tokens fail to distinguish malicious from legitimate activity, producing critical false negatives.

The key discriminant between the two scenarios lies in the structural execution pattern preserved in the provenance graph. The attack exhibits a 1:N "hub-and-spoke" topology—one compromised engineering host mediates commands to many field devices—whereas the maintenance task typically exhibits a 1:1 linear path between the engineer and a target PLC. Our framework leverages this structural signal (in addition to temporal cues) to disambiguate malicious misuse of legitimate tools.

2.1.2 Blind Spot 2: The Failure to See Causality in Structures. A graph-based analysis, which excels at modeling the holistic topology of system interactions, can fail when it lacks temporal and causal context. This is because benign administrative actions, when viewed structurally without considering the time difference between them, can appear identical to a malicious attack chain.

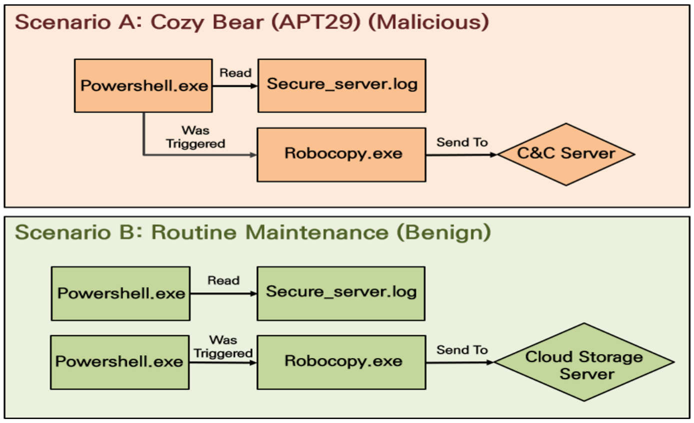

Figure 2: Limitation of Graph-Based Analysis - Malicious and Benign Behavioral Paths That Share Similar Structural Components

Consider the scenarios depicted in Figure 2. In a malicious Cozy Bear (APT29) attack (Scenario A), an adversary uses Powershell.exe to first read a sensitive file like Secure_server.log and then immediately triggers Robocopy.exe to exfiltrate the data as part of a single, continuous operation. In contrast, a benign routine maintenance setting (Scenario B) might involve the same actions as two separate, asynchronous events: an administrator could read the log for monitoring and, perhaps hours later, use Powershell.exe again to trigger Robocopy.exe for a scheduled backup.

From a purely structural perspective, the relationships are the same in both scenarios: the Powershell.exe node is connected to both the Secure_server.log and Robocopy.exe nodes. A graph-based model that only considers these relationships without their crucial temporal context would find it difficult to distinguish the attack from normal operations. The critical indicator of malice is the immediate, causal sequence of the actions—a temporal property that is lost in a purely structural view, making the threat invisible.

2.2 Enhancing Threat Detection through Cross-Modal Fusion

The necessity of a dual-modal approach is evident in how sophisticated attacks conceal their intent. In the TRISIS/Triton attack, for example, the malicious event sequence is identical to a benign maintenance task. The true anomaly is revealed only in the graph structure: a 1:N "hub-and-spoke" topology that indicates a coordinated, parallel attack, in stark contrast to the 1:1 linear path of the legitimate operation (Figure 3).

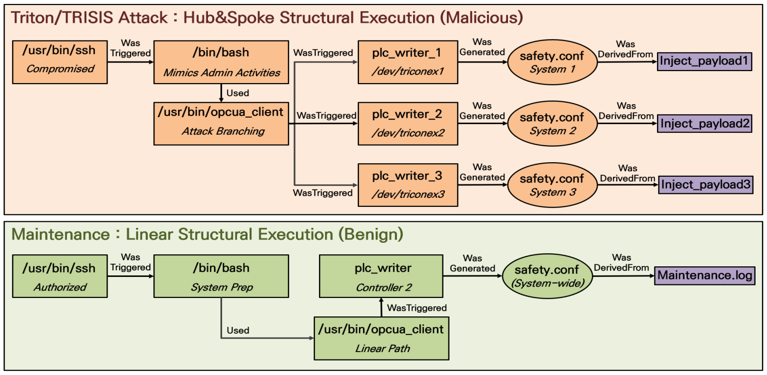

Figure 3: Same Causal Sequences, Different Structural Execution - 1:N Branching Attack Pattern vs. 1:1 Linear Maintenance Pattern

Conversely, in a Cozy Bear-style attack, the graph structure can mimic a routine backup operation. Here, the malicious indicator lies in the event sequence: the specific, temporally-bound chain of semantically suspicious actions where data reconnaissance (Read a sensitive file) is immediately followed by data exfiltration (Robocopy.exe) (Figure 4). These cases reveal that since threats can hide their intent in either the sequence or the structure, a robust detection system must jointly analyze both to achieve a complete understanding of system behavior. This principle is the foundational motivation for the propsed framework.

3 Related Work

Integrating of provenance analysis and deep learning has led to a paradigm shift in cyber threat detection. Prior studies in this domain can be broadly categorized into sequence-based and graph-based methods. Each has contributed significantly to understanding and detecting malicious system behavior, yet both exhibit inherent limitations that constrain their effectiveness in real-world, complex environments.

3.1 Sequence-Based Threat Analysis

Sequence-based approaches analyze system behaviors as temporally ordered events, assuming cyberattacks follow a distinct procedural flow. DeepLog [9], for example, applied LSTMs to system logs to model execution paths and detect malicious deviations by capturing short-term dependencies. However, this approach is inherently limited in understanding the broader structural context of how entities interact across different subsystems.

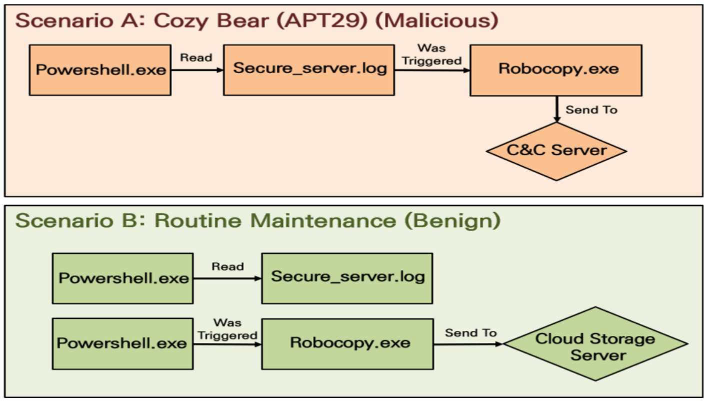

Figure 4: Similar Structural Components, Different Causal Sequences - Continuous Attack Path vs. Disconnected Benign Operations

To address data scarcity, DeepCASE [10] used a semi-supervised GRU-based framework to learn contextual representations from SIEM events, reducing the analyst's annotation workload. By operating only at the high-level SIEM event layer, this approach often misses the subtle, low-level nuances indicative of sophisticated attacks.

Recent studies have focused on detecting "LotL" attacks by using pattern-matching on command-line inputs, but their narrow focus limits their applicability to broader threat detection [11, 12].

Other systems, like ATLAS [13], apply LSTMs to construct attack timelines from provenance data but are designed for post-incident investigation and rely on prior alerts, limiting their use in real-time detection. Similarly, an LSTM-based approach by Villarreal–Vasquez et al. [14] for insider threat detection was tailored to a specific organizational context, lacking broader generalizability.

More recently, ConGraph [15] sought to enhance sequence-based models by incorporating contextual provenance information using a hybrid CNN and BiLSTM architecture. Although this improves accuracy, the fundamental process of flattening the provenance graph into a linear sequence results in the loss of vital structural details necessary to identify complex multi-stage attacks like lateral movement.

Despite their strength in modeling temporal sequences, these methods frequently fail to account for relationships spanning multiple subsystems, making them vulnerable to attacks that mimic legitimate sequences yet effect subtle structural manipulation.

3.2 Graph-Based Threat Analysis

Graph-based approaches model system activities as nodes and edges in a provenance graph, where nodes represent entities (processes, files) and edges encode causal relationships. This approach supports system-wide analysis, effectively capturing multi-entity relationships and lateral movement.

Among foundational contributions, ProvDetector [16] identifies threats by evaluating the rarity of event patterns, an assumption that often fails in dynamic contexts and causes high false positives. UNICORN [17] extends this for runtime APT detection but struggles with generalizability in heterogeneous environments.

More recent research has employed GNNs to learn directly from provenance graph topology [18, 19, 20]. While these techniques have advanced threat localization, they often rely on large labeled datasets, are computationally demanding, and typically lack mechanisms to model the temporal progression of staged attacks.

To improve efficiency, sketch-based methods like StreamSpot [21] and HistoSketch [22] offer high scalability by decomposing graphs into fixed-size structures, but this comes at the cost of losing critical contextual and semantic information.

Hybrid approaches attempt to combine insights. PROGRAPHER [23] incorporates coarse-grained features into graph embeddings, while EagleEye [24] applies transformers to event streams but treats sequence and graph structures as separate modalities, missing opportunities to exploit the mutual reinforcement between temporal and structural signals.

Recent large language model (LLM)-based models like OMNISEC [25] and PROVSYN [26] excel at fine-grained semantic analysis and synthetic data generation, respectively. However, both focus solely on structural patterns and lack the explicit temporal modeling needed to detect complex multi-stage attacks.

3.3 Limitations of Existing Approaches

Despite significant advancements, current threat detection approaches exhibit limitations that hinder their effectiveness in complex operational environments such as industrial control and cyber-physical systems (CPS).

Single-modality learning faces a fundamental limitation: while sequence-based methods [9-15] model causal event flows effectively, they overlook the structural context of entity relationships. Conversely, graph-based models [16-26] excel at modeling structural dependencies but overlook the temporal dynamics crucial for understanding the staging and escalation of multi-step attacks. These modality-specific blind spots leave systems vulnerable to stealthy adversaries who craft attacks to appear benign when viewed via any single perspective.

Moreover, many existing approaches [16,21,22] depend on coarse-grained features—such as entity types or event categories—which lack the semantic depth required to distinguish between legitimate complex operations (e.g., system updates or backups) from malicious behaviors with similar surface patterns. This limitation is especially problematic in operational environments where normal behavior is diverse and context-dependent.

While multi-modal or hybrid methods have been introduced [23,24], they commonly rely on simplistic fusion mechanisms, such as parallel processing streams or direct feature concatenation, which offer limited interaction between modalities. This approach restricts the model's ability to align temporal dynamics with structural anomalies effectively, ultimately compromising detection precision.

Finally, a majority of provenance-based systems [17,18,19,25,26] are tailored to either information technology (IT)-centric infrastructures or graph-centric pipelines requiring fully constructed graphs before inference. Such constraints undermine their applicability to dynamic or real-time operational settings—particularly in OT or ICS environments—where cross-domain generalizability and low-latency inference are essential.

These limitations collectively underscore the need for a cross-modal architecture that jointly captures both temporal progression and structural relationships, while preserving semantic richness and generalizability across domains. In response, we propose a deep learning framework that unifies sequential and graph-based representations to enable robust and context-aware threat detection in both IT and OT systems.

Table 1 presents a structured summary of representative approaches [9-26] to further clarify the comparative landscape and the innovation embodied in the proposed model. The comparison spans their core modalities, modeling capabilities, semantic expressiveness, integration strategies, and target domains.

Prior research in provenance-based threat detection has largely been bifurcated, evolving along two distinct methodological trajectories focusing on either sequential or structural analysis. This separation stems not from an oversight, but from substantial and practical challenges. A primary factor is the inherent architectural complexity involved in fusing heterogeneous data structures. Integrating the ordered, temporal dynamics of event sequences with the unordered, topological properties of graphs demands sophisticated mechanisms for latent space alignment, a challenge that has only become tractable with recent advancements in deep learning, such as attention-based fusion.

Model / Work	Temporal Modeling	Structural Modeling	Semantic Feature Richness	Cross-Modal Fusion	Application Domain
DeepLog[9]	✓	✗	low	✗	IT
DeepCASE[10]	✓	✗	Medium	✗	IT
Ongun et al.[11]	✓	✗	Low	✗	IT
ATLAS[13]	✓	✗	Medium	✗	IT
ConGraph[15]	✓	✓	Medium	✗	OT
ProvDetector[16]	✗	✓	Low	✗	IT
UNICORN[17]	✗	✓	Low	✗	IT
SIGL[18]	✗	✓	Medium	✗	IT
Shadewatcher [19]	✗	✓	Medium	✗	IT
ThreaTrace[20]	✗	✓	Medium	✗	IT
StreamSpot[21]	✗	✓	Low	✗	IT
HistoSketch[22]	✗	✓	Low	✗	IT
Prographer[23]	✗	✓	Medium	✗	IT
EagleEye[24]	✓	✗	High	✗	IT
OMNISEC[25]	✗	✓	High	✗	IT
PROVSYN[26]	✗	✓	High	✗	IT
Proposed	✓	✓	**High**	✓	**IT/OT**

Table 1: Comparative Analysis of Threat Detection Models

Furthermore, the siloed evolution of these research domains, each with its own set of benchmarks and community-specific problems, has historically discouraged the development of hybrid methodologies. Compounding this is the significant computational overhead of dual-branch models, which often presents a practical barrier, incentivizing research to first exhaust the capabilities of more resource-efficient, single-modality frameworks. Consequently, this has created a persistent modality gap. Proposed model is designed to explicitly bridge this gap, leveraging mature deep learning techniques to synthesize temporal and structural insights into a single, cohesive detection framework.

4 The Cross-Modal Detection Framework

The proposed cross-modal deep learning framework resolves key shortcomings in provenance-based threat detection. Figure 5 presents the comprehensive system overview, illustrating the complete pipeline from raw system logs to final threat classification in ICSs. The architecture comprises four core stages: (1) provenance graph generation, (2) security feature extraction and embedding, (3) behavioral modeling, and (4) threat detection. Unlike existing single-modality approaches that suffer from information loss, this framework simultaneously processes both temporal sequences and graph structures, enabling a comprehensive understanding of complex attack scenarios.

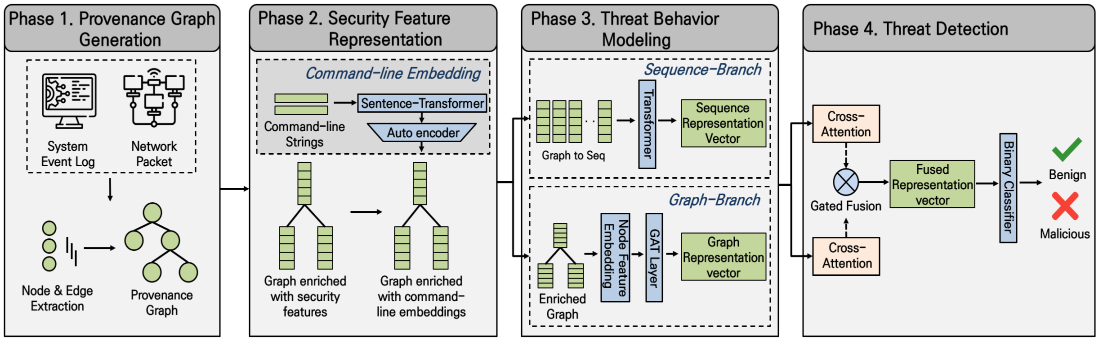

Figure 5: Overview of the Proposed System Architecture

4.1 Provenance Graph Generation

Figure 6 presents the provenance graph generation pipeline, outlining temporal window segmentation, entity and relationship extraction, and graph structure representation steps that process raw audit logs into structured provenance graphs optimized for cross-modal analysis.

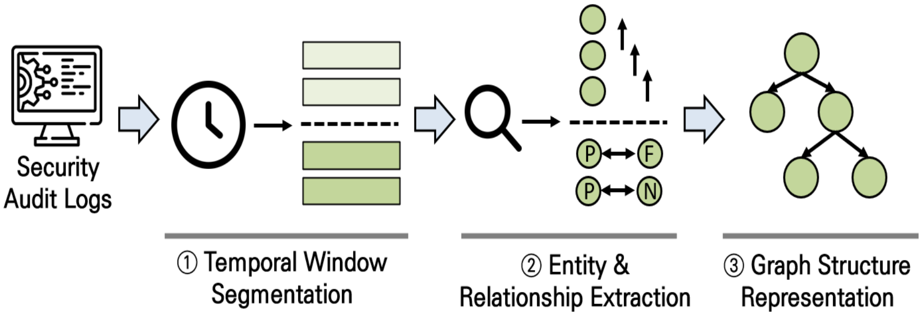
Security Audit Logs
P
F
P
N
① Temporal Window Segmentation
② Entity & Relationship Extraction
③ Graph Structure Representation

Figure 6: Provenance graph generation pipeline

4.1.1 Temporal Window Segmentation. Our framework employs a strategic temporal windowing approach to address scalability challenges inherent in large-scale provenance graphs. The window size is determined by analyzing typical attack pattern durations and computational requirements, providing an optimal balance between contextual completeness and computational tractability while preserving vital attack chain information.

This segmentation strategy not only improves analytical focus but also ensures computational feasibility in large-scale ICS deployments. By limiting graph size within each window and performing independent inference, the system achieves linear scalability with respect to time and maintains consistent latency even under continuous data ingestion. This design consideration supports the real-time operational requirements typical of industrial monitoring environments.

The window selection process considers multiple factors: (1) Attack Pattern Analysis: Multi-stage attack scenarios require sufficient temporal context to capture complete attack progressions from initial compromise to objective completion; (2) Computational Efficiency: Window sizes must maintain processing requirements within available computational resources while enabling real-time analysis; (3) Contextual Preservation: Temporal intervals must provide adequate context for elucidating causal relationships between system events without excessive computational overhead.

The temporal window size was determined through empirical analysis of multi-stage attack sequences and operational constraints observed in industrial control environments. Across representative attack scenarios, the typical duration from initial compromise to impact completion ranges between 10 and 20 minutes. This observation guided the selection of a 15-minute window as a balanced temporal context for capturing complete causal chains of activity while maintaining computational tractability.

In addition to aligning with observed attack dynamics, this duration corresponds to common operational cycles in industrial systems, such as periodic sensor polling, process control loops, and safety monitoring intervals, which typically operate on a 10–20 minute cadence. Hence, a 15-minute fixed segmentation provides a domain-consistent and operationally meaningful unit for behavioral modeling.

It is important to note that the proposed framework does not rely on a fixed parameter: the window size can be dynamically adjusted during deployment depending on log density, system throughput, and real-time monitoring requirements. This flexibility ensures that the temporal segmentation remains adaptable and generalizable across different industrial and enterprise contexts.

4.1.2 Entity and Relationship Extraction. Within each temporal window, the system extracts three primary entity types following the W3C PROV (Provenance) standard: Processes characterized by process ID, executable file path, and command-line arguments; Files identified by file path, size, permissions, and cryptographic hash values; and Network connections defined by IP addresses, port numbers, and communication protocols.

The system establishes five fundamental relationship types that capture the essential causality patterns in system behavior: WasGeneratedBy relationships indicate files created by specific processes; Used relationships represent process access to files or system resources; WasTriggeredBy relationships capture process spawning and execution chains; WasInformedBy relationships model inter-process communication and coordination; and WasDerivedFrom relationships track file derivation and modification patterns.

4.1.3 Graph Structure Representation. The constructed provenance graphs form DAGs where nodes represent system entities and edges capture causal relationships with temporal ordering. Each graph maintains local structural properties (immediate neighbor relationships) and global topological characteristics (centrality measures, clustering coefficients, path lengths) essential for comprehensive security analysis.

The graph representation preserves critical information including temporal causality via timestamped edges that maintain the chronological order of system events; hierarchical relationships such as process parent-child structures and file derivation chains; cross-domain interactions between processes, files, and network activities that characterize complex attack patterns; and structural signatures that distinguish normal system operations from malicious activities based on graph topology.

4.2 Security Feature Representation

Figure 7 demonstrates the security feature representation process, highlighting the construction of 156-dimensional feature vectors through multi-dimensional security feature engineering, contextual feature enrichment, and command-line embedding techniques.

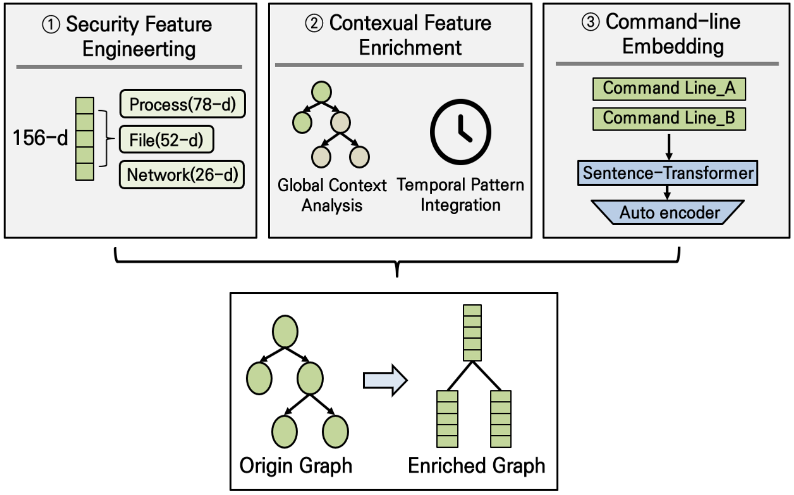

Figure 7: Security feature representation process

4.2.1 Multi-Dimensional Security Feature Engineering. The system performs security feature extraction by converting raw graph entities into 156-dimensional vectors tailored for threat detection in industrial control systems.

Process features (78 dimensions) combine basic attributes (e.g., pid, command_line) with computed behavioral indicators. These include statistical analysis of system call frequencies to detect privilege escalation, network metrics derived from connection counts, file system interaction patterns like access frequency, and temporal patterns that identify off-hours or burst activities.

File features (52 dimensions) integrate direct attributes (e.g., file_path, permissions) with computed risk indicators. Key indicators include entropy scoring to detect data staging or obfuscation, path-based risk assessment that flags known suspicious locations (e.g., /tmp/, /dev/shm/), and access pattern analysis from provenance graph relationships.

Network features (26 dimensions) enhance basic connection attributes (e.g., destination_ip, port) with computed risk metrics. These metrics are derived from connection pattern analysis for DGA detection, protocol analysis to identify suspicious usage, and traffic volume analysis to detect abnormal upload/download ratios.

Table 2 summarizes the feature categories and provides representative examples of the 156-dimensional security feature vector, demonstrating the comprehensive coverage across process behaviors, file characteristics, and network activities essential for robust threat detection in ICS environments.

Table 2: Security Feature Engineering: 156-Dimensional Feature Categories

Class	Count	Representative Features

Process Features	78	process_name, command_line, setuid_call_frequency, external_connection_count, off_hours_activity
File Features	52	file_path, file_size, shannon_entropy_score, path_risk_assessment, multi_process_access_count
Network Features	26	destination_ip, protocol, external_ip_entropy, dga_pattern_score, upload_download_ratio

Each feature is derived from a predefined set of statistical, structural, and security-domain indicators. For example, statistical metrics such as activity entropy or temporal concentration quantify the variability of event distributions within a window, while structural metrics such as graph density or clustering-coefficient change capture the connectivity and evolution of the provenance subgraph. Security-related indicators, including process frequency deviation and privilege escalation flags, represent behavioral and contextual deviations. All features are computed automatically through a unified preprocessing pipeline and normalized before model training.

4.2.2 Contextual Feature Enrichment. Beyond individual entity features, the framework employs global context analysis to enrich features with graph-wide information. This involves analyzing an entity's role within the entire graph (e.g., as a communication hub) and identifying suspicious cross-entity correlation patterns, such as a process accessing a sensitive file before initiating a suspicious network connection.

Temporal pattern integration adds time-based behavioral analysis—considering the timing, frequency, and sequencing of activities—to identify automated behaviors typical of malware. This contextual enrichment is crucial for detecting sophisticated attacks that rely on subtle coordination between multiple system components.

4.2.3 Command-Line Embedding. The system incorporates a novel command-line embedding mechanism to capture semantic relationships in commands, enabling the detection of sophisticated "living-off-the-land" techniques. By processing both current and parent process command lines, it captures hierarchical execution context to trace complex attack chains distributed across multiple process levels.

The embedding process uses a pre-trained sentence transformer model (all_MiniLM_L6_v2) to generate 384-dimensional semantic embeddings. For real-time efficiency, an autoencoder then compresses these into 16-dimensional vectors, retaining semantic integrity while minimizing computational overhead.

These compressed embeddings are integrated into the broader 156-dimensional feature vector, providing rich contextual information that complements conventional security indicators and helps detect attacks relying on command-line manipulation.

4.3 Threat Behavior Modeling

4.3.1 Dual-Branch Architecture Design. The core innovation is a dual-branch architecture that simultaneously processes provenance data as both temporal sequences and graph structures. This dual approach prevents the information loss inherent in single-modality methods, ensuring a comprehensive capture of both temporal and structural attack patterns. Figure 8 illustrates this behavioral modeling component.

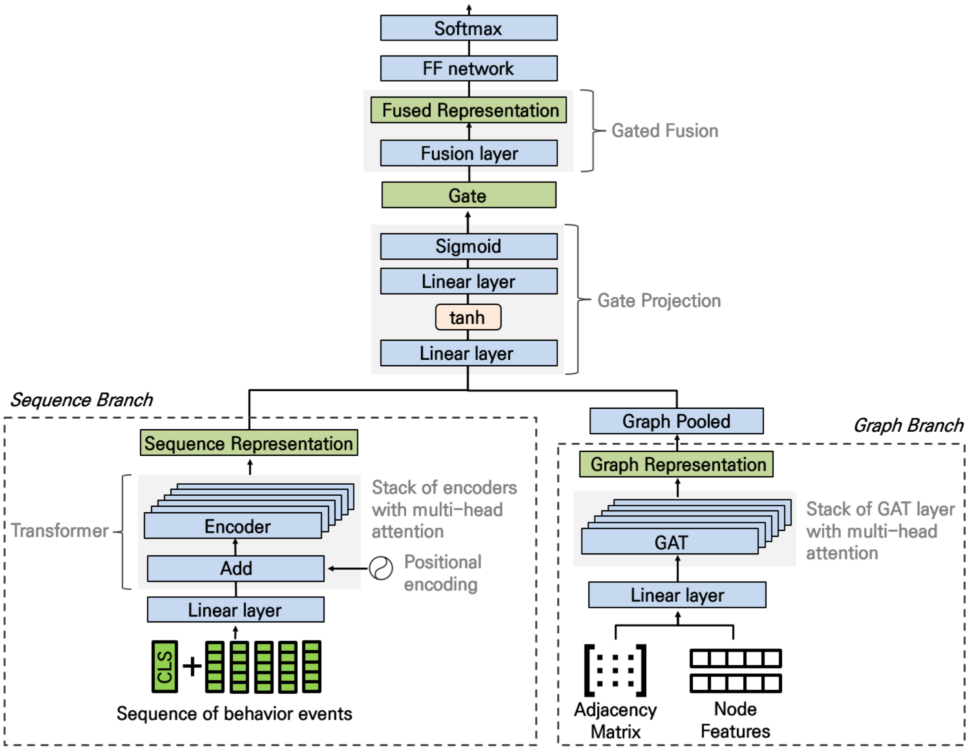

Figure 8: Behavior Modeling-Based Dual-Branch Architecture

The sequence branch transforms graphs into ordered sequences for transformer-based analysis of multi-stage attacks, while the graph branch processes the complete structure to capture global topological patterns. Though operating on the same data, they extract complementary security insights for a comprehensive behavioral understanding.

4.3.2 Sequence Branch: Temporal Behavioral Pattern Learning. The sequence branch uses graph-to-sequence transformation to capture the temporal dynamics and causal event order of multi-stage attacks.

Unlike methods that simply flatten events, our framework uses intelligent graph traversal algorithms (Depth First Search (DFS)/Breadth First Search (BFS)) to extract meaningful causal pathways. These traversals follow causal edges (e.g., WasTriggeredBy, Used) to maintain semantic coherence. The system then uses a comprehensive vocabulary of 2,000 security-relevant tokens that combine structural and temporal semantics.

The branch employs a 6-layer transformer encoder (8 attention heads, 512-dim embeddings) optimized for graph-derived sequences. Key features include path-aware token embedding, positional encoding for temporal order, and causality-aware attention. This architecture allows the model to capture long-range dependencies critical for detecting sophisticated APTs. Notably, the transformer was also chosen for its interpretability, as visualizing attention weights can reveal which events were most influential in a malicious classification.

4.3.3 Graph Branch: Structural Behavioral Pattern Learning. While the sequence branch analyzes individual causal paths, the graph branch examines the complete graph structure to capture global patterns missed by path-based analysis. This enables the detection of complex attacks like lateral movement and coordinated campaigns.

By processing the graph as a unified structure, it can detect centrality anomalies, malicious subgraph patterns, influence propagation, and other structural signatures that distinguish malicious from benign operations.

The branch employs a 6-layer GAT with 4 attention heads. Its multi-head attention mechanism allows each head to focus on different structural relationships, while dynamic weighting assigns higher importance to suspicious graph regions. This multi-layer architecture enables the multi-hop information propagation necessary to capture attack patterns spanning several degrees of separation. The GAT was also chosen for its explainability, as its attention scores can highlight the critical nodes and edges that contributed most to a detection.

4.4 Threat Detection

4.4.1 Cross-Modal Fusion Mechanism. The framework's most significant innovation is its cross-modal fusion mechanism, detailed in Figure 8. A cross-modal attention module enables bidirectional information exchange between sequence and graph representations, achieving a completeness that neither modality can provide alone. Specifically, sequence-to-graph attention allows temporal patterns to guide structural analysis, while graph-to-sequence attention allows structural patterns to inform temporal analysis.

Aligning these modalities enhances detection: temporal patterns establish causal order, while structural patterns reveal global dependencies. The mechanism also dynamically adjusts the importance of each modality based on the detected patterns, emphasizing sequence information for multi-stage attacks and structural information for coordinated, multi-target operations.

4.4.2 Gated Fusion Architecture. The framework uses a gated fusion mechanism, formulated as α=sigmoid(W[seq_repr;graph_repr]), to dynamically balance the contributions of temporal and structural information. In this formulation, seq_repr and graph_repr denote the feature representations obtained from the sequence and graph branches, respectively; W is a learnable transformation matrix; and the sigmoid function σ(·) produces the gating coefficient $\alpha \in [0, 1]$, which controls the relative influence of each modality.

This adaptive gating learns optimal integration during training, emphasizing sequence information for temporally evolving APTs and graph information for distributed structural attacks. Through mutual enrichment, temporal context enhances tructural analysis while structural context refines sequential reasoning, achieving comprehensive representation learning. Moreover, the learned gate values improve interpretability by indicating whether each detection decision is primarily driven by temporal or structural cues.

4.4.3 Threat Classification. The final stage uses a multi-layer classification head to produce binary threat assessments from the fused representations. The architecture incorporates domain-specific needs for critical infrastructure, such as asymmetric cost modeling that prioritizes threat detection over minimizing false positives.

Binary Classification Framework: The framework is designed for binary classification because in ICS environments, the cost of a missed threat (false negative) far exceeds the cost of a false positive. The classification head uses multiple regularized dense layers to maintain discriminative capability for subtle attack indicators without overfitting.

5 Evaluation

5.1 Datasets

Our evaluation employs the CICAPT-IIoT dataset [27], a comprehensive cybersecurity dataset for IIoT/ICS environments. Unlike conventional network-centric data, it provides system-level provenance traces that capture complete causal chains of both benign activities and sophisticated attack scenarios.

The dataset contains ~100,000 provenance nodes collected over 7 days from a sophisticated hybrid testbed. This testbed integrates physical devices (e.g., Raspberry Pi controllers) and virtual components running industrial protocols (MODBUS, MQTT) to simulate realistic scenarios like power grid management and water treatment systems.

The attack scenarios are based on APT29 (Cozy Bear) Tactics, Techniques, and Procedures (TTPs) adapted for Linux environments, covering over 20 techniques across major tactics like collection, exfiltration, and lateral movement. Each attack uses a characteristic "low and slow" approach, with steps executed at random 45-75 minute intervals to simulate realistic stealth patterns.

To address the inherent class imbalance of cybersecurity data, we bypassed conventional oversampling in favor of an advanced graph construction and augmentation strategy. Our approach segments the continuous provenance stream into 15-minute temporal windows, each forming a complete subgraph. From each window, we extract multiple, diverse provenance graphs using different traversal strategies, ensuring robust and semantically consistent samples for model training.

This strategy generated 47,832 high-quality provenance traces with a balanced distribution of 26,307 benign (55%) and 21,525 malicious (45%) samples. After preprocessing, the average trace contains 178 nodes and 423 edges, with sequences averaging 347 events. This composition provides a realistic and trainable distribution for comprehensive model evaluation.

5.2 Evaluation Setup

5.2.1 Cross-Modal Data Generation. The framework generates corresponding inputs for both of its architectural branches. The sequence generation employs graph traversal algorithms (DFS/BFS) to extract semantically meaningful causal pathways, ensuring logical and temporal coherence. This graph-to-sequence conversion prioritizes security-relevant paths and extracts multiple sequences from each graph (up to 512 tokens long) to balance context with computational efficiency.

The graph branch receives the complete provenance graph structure. Preprocessing includes node feature normalization, adjacency matrix construction with edge type encoding, and the addition of self-loops to handle isolated nodes while maintaining the original topology.

The dataset is split into training (70%), validation (15%), and test (15%) sets using stratified random sampling to maintain class balance. To prevent data leakage, the partitioning strategy ensures temporal independence by clustering traces from the same time periods and assigning entire clusters to a single partition. Complex attack campaigns spanning multiple traces are also kept within one partition to prevent artificially inflated performance metrics. The resulting dataset composition is summarized in Table 3.

Table 3: CICAPT-IIoT Dataset Distribution: Training, Validation, and Test Set Composition

Dataset Class	Benign	Malicious	Total
Train	18,415	15,067	33,482 (70%)
Validation	3,949	3,226	7,175 (15%)
Test	3,943	3,232	7,175 (15%)

5.2.2 Hyperparameter Configuration. The model's hyperparameters were optimized via systematic grid search. Both sequence and graph branches use 512-dimensional embeddings and 6 encoder layers.

The sequence branch employs 8 attention heads and the graph branch 4 heads. A learning rate of 2e-4 with a cosine annealing scheduler and a batch size of 16 were used for stable convergence. Training ran for a maximum of 50 epochs with early stopping (patience=20) to prevent overfitting.

To address class imbalance, a Focal Loss configuration of α=0.69 and γ=1.65 was used to emphasize difficult-to-classify samples. The classification threshold was optimized to 0.548 to maximize the F1-score. Additionally, mixed precision training was employed to reduce memory consumption by 50%, enabling larger batch sizes on standard GPU hardware.

5.2.3 Evaluation Metrics. Our evaluation employs metrics relevant to ICS security operations, where false positives and false negatives have different operational costs. Precision is crucial to avoid alert fatigue and maintain operator responsiveness to legitimate threats.

Recall is essential to prevent catastrophic incidents from missed attacks. The F1-score provides a balanced assessment of the precision-recall trade-off, which is critical given the significant costs of both false positives and negatives in industrial security.

5.3 Results & Discussion

5.3.1 Performance Comparison with Baseline Methods. To comprehensively evaluate our proposed framework, we benchmarked it against a carefully selected spectrum of baseline models, with the rationale of establishing performance across different levels of complexity.

We included conventional machine learning methods (Random Forest (RF), Support Vector Machine (SVM)) and basic neural networks to demonstrate the limitations of threat detection based on shallow feature learning. The most critical comparison, however, was against state-of-the-art single-modality deep learning models, such as a Transformer for sequence-based analysis and a GAT for structural analysis. This direct comparison was designed to empirically test our central hypothesis that a cross-modal fusion approach surpasses the performance of even the most advanced single-modality methods by addressing their fundamental blind spots.

This study evaluates the proposed framework against fundamental baseline methods to establish the effectiveness of cross-modal deep learning approaches for provenance-based threat detection. Table 4 presents comprehensive performance metrics, while Figure 9 illustrates the corresponding receiver operating characteristic (ROC) curve analysis, demonstrating superior performance characteristics across all evaluation criteria.

Table 4: Performance Comparison of Framework Against Baseline Models

Category	Method	Accuracy	Precision	Recall	F1 score
Conventional ML	RF	78.32%	72.17%	85.23%	78.1%
	SVM	80.16%	74.3%	83.73%	78.74%
Basic Neural Networks	MLP	76.81%	70.54%	82.41%	76.01%
	Basic CNN	79.24%	73.82%	84.1%	78.62%
Sequence-Based	LSTM	82.71%	76.81%	88.35%	82.18%
	Transformer	87.26%	84.12%	86.98%	85.5%
Graph-Based	Graph Convolution Network (GCN)	84.28%	79.6%	86.11%	82.76%
	GAT	85.94%	81.46%	87.62%	84.4%
Proposed	**GAT + Transformer**	**93.96%**	**95.31%**	**91.04%**	**93.13%**

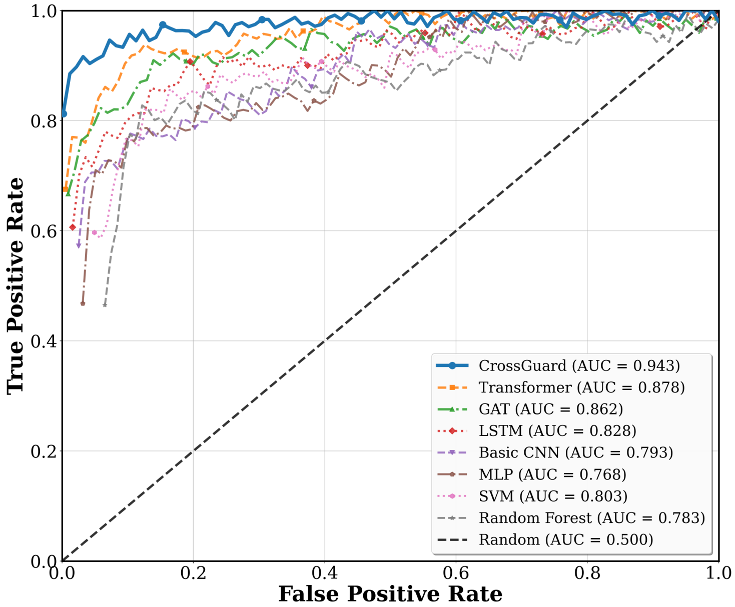

Figure 9: ROC Curves: Baseline Methods Comparison

Conventional machine learning approaches demonstrate inherent limitations in capturing complex behavioral patterns. Random Forest achieves 78.32% accuracy with an area under curve (AUC) of 0.783, while SVM achieves 80.16% accuracy, indicating that manual feature engineering is insufficient for modeling sophisticated threat signatures. Basic neural networks yield limited improvements, with multi-layer perceptron (MLP) recording 76.81% accuracy and Basic Convolutional Neural Network (CNN) achieving 79.24% accuracy.

Graph-extracted causal pathway approaches validate the importance of causality modeling. LSTM achieves 82.71% accuracy with an AUC of 0.828, while the transformer exhibits significant advancement at 87.26% accuracy with an AUC of 0.878. Graph-based methods demonstrate competitive performance, with GAT achieving 85.94% accuracy and an AUC of 0.862, validating the importance of structural information in threat detection.

Our model achieves exceptional performance across all metrics, attaining 93.96% accuracy, 95.31% precision, and an AUC of 0.943. Most significantly, our method maintains true positive rates above 0.85 in the critical low-FPR region. The comprehensive evaluation demonstrates that the framework's cross-modal architecture effectively addresses single-modality limitations through synergistic fusion of graph-extracted causal pathways and direct structural analysis, achieving a +6.70% accuracy improvement over the best baseline with a +11.19% precision enhancement.

5.3.2 Comparison with State-of-the-Art Methods. This study benchmarks our proposed framework against recent state-of-the-art provenance-based threat detection models from leading security conferences and journals to highlight its advancements. Table 5 presents comprehensive evaluation results, while Figure 10 illustrates the corresponding ROC analysis demonstrating superior classification performance across all operating points.

Table 5: Performance Comparison of Framework against State-of-the-Art Models

Category	Method	Accuracy	Precision	Recall	F1 score
Graph-Based	EagleEye[8]	91.2%	89.5%	92.1%	89.6%
	ProvDetector[16]	89.7%	87.3%	88.7%	89.1%
	UNICORN[17]	88.4%	86.7%	89.2%	87.9%

Sequence-Based	DeepCase[10]	87.8%	86.2%	89.3%	87.7%
	DeepLog[9]	85.4%	83.6%	87.8%	85.6%
Proposed	**GAT + Transformer**	**93.96%**	**95.31%**	**91.04%**	**93.13%**

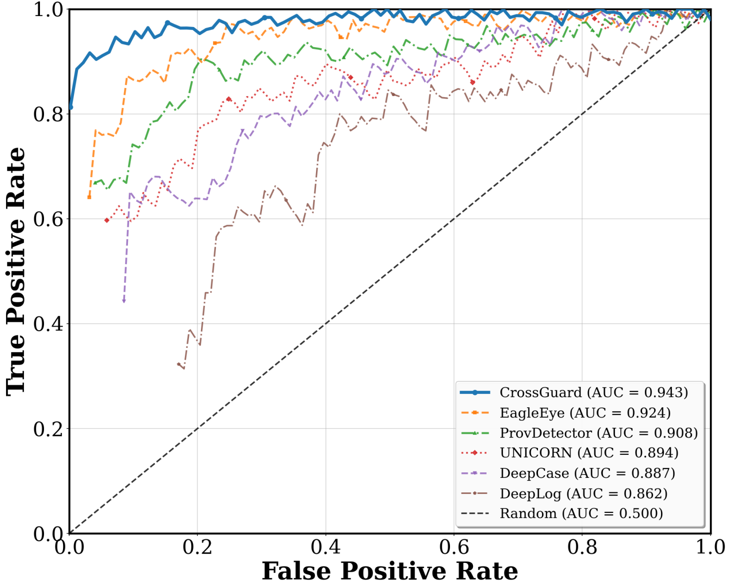

Figure 10: ROC Curves: State-of-the-Art Methods Comparison

EagleEye represents the current best-performing approach with 91.2% accuracy and 89.6% F1-score among graph-based methods. Our model outperforms EagleEye by +2.76% in accuracy and +3.53% in F1-score, achieving an AUC of 0.943 compared to EagleEye's 0.924. ProvDetector (89.7% accuracy, AUC 0.908) and UNICORN (88.4% accuracy, AUC 0.894) exhibit larger performance gaps, with our model demonstrating +4.26% and +5.56% accuracy improvements respectively.

Sequence-based methods reveal substantial performance gaps highlighting the limitations of purely sequential approaches. DeepCase achieves 87.8% accuracy with an AUC of 0.887, while SLEUTH reaches 85.4% with an AUC of 0.862. Our model outperforms these methods by +6.16% and +8.56% in accuracy respectively, with particularly notable precision improvements of +9.11% over DeepCase.

The comprehensive evaluation demonstrates that our framework's cross-modal architecture effectively addresses the fundamental limitations of single-modality approaches. Most significantly, our model's exceptional performance in the low-false positive rate (FPR) region (0.0-0.1), where it maintains true positive rates above 0.85, validates its suitability for security operations centers. By uniquely combining graph-extracted causal pathway analysis with direct structural modeling, our approach achieves state-of-the-art performance while maintaining computational efficiency suitable for real-world deployment.

5.3.3 Cross-Modal Architecture Validation through Ablation Study. To dissect and validate the contribution of each architectural component, we conducted a systematic ablation study. This study was designed to progressively build our final model, allowing us to quantify the performance gain at each stage. The logic was to first establish the independent performance of the sequence and graph branches, then to show the modest benefit of a naive concatenation, and finally to introduce our core innovation—the cross-modal attention mechanism. This progressive analysis allows us to precisely isolate and measure the performance uplift attributable to the synergistic, bidirectional information exchange between temporal and structural modalities, thereby validating our key design decisions.

Table 6 presents the quantitative contribution of each design decision, while Figure 11 illustrates the corresponding ROC curves demonstrating progressive improvement. To ensure statistical reliability, each configuration was evaluated over five independent runs with different random seeds, and the mean ± standard deviation values are reported. To ensure consistency, each experiment was repeated with stratified random sampling across training, validation, and test sets. The performance deviation remained within ±0.4%, confirming that the reported improvements are statistically stable and reproducible.

Table 6: Ablation Study of Framework Components and Their Impact on Detection Performance

Config	Accuracy	Precision	Recall	F1 score
Sequence Branch Only	87.2% ± 0.3%	84.1% ± 0.4%	86.9% ± 0.2%	85.5% ± 0.3%
Graph Branch Only	85.9% ± 0.4%	81.4% ± 0.3%	87.6% ± 0.4%	84.4% ± 0.3%
Simple Concatenation	89.1% ± 0.3%	86.3% ± 0.3%	88.2% ± 0.2%	87.4% ± 0.3%
+ Cross-Modal Attention	91.3% ± 0.2%	89.7% ± 0.3%	89.5% ± 0.2%	89.6% ± 0.3%
+ Focal Loss	92.8% ± 0.2%	93.2% ± 0.3%	90.1% ± 0.3%	91.6% ± 0.2%
+ Threshold Optimization	93.96% ± 0.3%	95.31% ± 0.3%	91.04% ± 0.3%	93.13% ± 0.2%

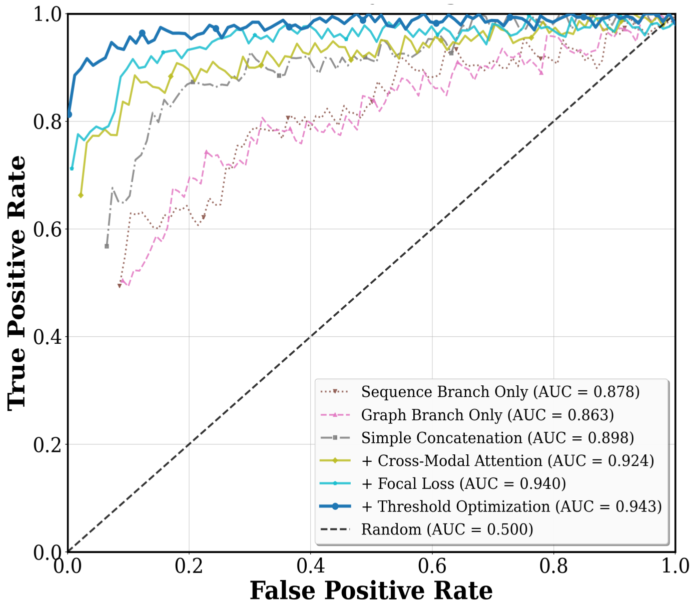

Figure 11: ROC Curves: Ablation Study Progressive Improvement

Single-modality baselines establish our cross-modal foundation. The sequence branch processing graph-extracted causal pathways achieves 87.2% accuracy with an AUC of 0.878, while the graph branch analyzing structural relationships achieves 85.9% accuracy with an AUC of 0.863. The sequence branch's slight advantage suggests that causal progression patterns provide marginally stronger discriminative signals for the CICAPT-IIoT dataset.

Simple concatenation improves performance to 89.1% accuracy with an AUC of 0.898, representing a modest 1.9% improvement over single-modality approaches. However, cross-modal attention mechanisms yield substantial gains, elevating accuracy to 91.3% with an AUC of 0.924. This 2.2% improvement validates the importance of bidirectional information exchange between modalities, with ROC analysis demonstrating marked improvement in the critical low-FPR region.

Focal loss incorporation produces significant precision improvements from 89.7% to 93.2% while maintaining recall, addressing class imbalance challenges inherent in cybersecurity datasets. Finally, data-driven threshold optimization provides ultimate enhancement, improving accuracy to 93.96% and precision to 95.31% by identifying 0.548 as the optimal threshold, departing from the conventional 0.5.

The comprehensive ablation study validates each component's contribution, with a cumulative 6.76% accuracy improvement from single-branch baseline to full architecture demonstrating the necessity of our cross-modal approach for state-of-the-art performance.

To complement the quantitative results in Table 6, we examined the nature of errors introduced when specific components were ablated. The single-modality branches reveal distinct weaknesses: the sequence-only model exhibits increased false negatives, particularly in multi-process coordination attacks (e.g., APT29 lateral movement), where structural dependencies are critical. In contrast, the graph-only model produces more false positives, often misclassifying routine administrative or maintenance operations that form dense but benign subgraph structures.

When naive concatenation was applied, errors persisted because inter-modal dependencies were treated independently, leading to misalignment between causal timing and structural context. The introduction of cross-modal attention significantly mitigated both failure types by reinforcing temporal-structural alignment, but some false negatives remained under extremely sparse activity intervals. Incorporating focal loss reduced sensitivity bias toward dominant benign classes, while threshold optimization effectively suppressed low-confidence false alarms.

Overall, the ablation error patterns confirm that each module contributes not only to quantitative improvement but also to error-type mitigation: cross-modal attention reduces false negatives, focal loss balances class distribution, and threshold calibration enhances operational precision.

5.3.4 Threat Detection Specificity and Operational Impact. The confusion matrix analysis, shown in Figure 12, demonstrates the model's strong classification performance. The model correctly identified 2,939 of 3,232 malicious traces, achieving a high sensitivity of 90.9%. Among the 3,943 benign traces, only 139 were misclassified, resulting in a low false positive rate of 3.5%.

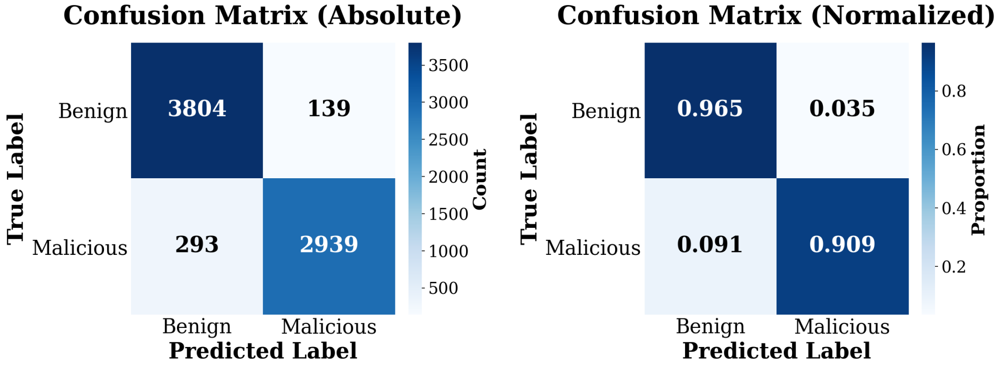

Figure 12: Confusion Matrix: Framework Performance on Test Dataset

We acknowledge that in an operational context, a 3.5% false positive rate—translating to approximately 20 alerts per day in this dataset—warrants careful consideration to avoid alert fatigue. However, while an absolute false positive rate presents operational challenges, it is crucial to note that this represents a 72% reduction compared to existing state-of-the-art methods. Our work significantly lowers the barrier to practical deployment by drastically reducing the number of false alarms an operator must investigate.

To this end, the proposed system is designed to function as a high-fidelity Tier-1 filtering system. Its purpose is to elevate a small number of highly suspicious events from millions of raw logs, rather than to eliminate the need for human analysis entirely. The ~20 daily alerts, while not insignificant, are far more manageable for a security operations center (SOC) than analyzing raw data streams.

Furthermore, in critical infrastructure environments, the cost of a missed attack (a false negative) can be catastrophic and far outweighs the operational cost of investigating a false positive. Our threshold optimization prioritizes maintaining a high detection rate while pushing false positives to an operationally tolerable level. By functioning as an intelligent alert prioritization engine, the system enables security teams to focus their limited resources on the most credible threats, aligning with the core safety and security principles of ICS operations.

5.3.5 Deployment Efficiency and Scalability Considerations. Although direct latency benchmarking under live ICS workloads was not performed, the framework was explicitly designed for deployment efficiency and scalable inference. The temporal window segmentation mechanism (15-minute fixed window) limits the number of nodes processed per batch, keeping the per-inference graph below 5k edges on average. This design ensures that computation remains bounded regardless of continuous log accumulation.

In practice, using mixed-precision training and batch normalization, inference on a single window takes approximately 0.8 seconds on an RTX 4090 GPU (batch size = 16), corresponding to near-real-time throughput for continuous monitoring. For CPU-only inference on an industrial workstation (Xeon Silver 4314), the latency per window remains under 4.5 seconds, sufficient for periodic analytics or offline correlation tasks.

The model architecture also supports horizontal scalability. Each temporal window is independent, allowing distributed inference across multiple GPUs or microservices. This enables integration with real-time provenance stream pipelines, such as Kafka-based log collectors, without compromising latency or detection accuracy.

Future work will empirically evaluate throughput on live plant testbeds to benchmark streaming efficiency, latency under burst conditions, and resource utilization metrics (CPU/GPU and memory).

6 Conclusion

In this study, we introduced a novel cross-modal deep learning framework designed to overcome the fundamental limitations of single-modality threat detection in industrial control systems. By synergistically fusing a transformer-based sequence branch with a GAT-based graph branch, our framework uniquely captures both the temporal progression and the structural context of system behaviors. Our experimental results validate the effectiveness of this approach, demonstrating superior detection accuracy (93.96%) and a significant reduction in false positives.

While the results are promising, we acknowledge certain limitations that present avenues for future work. Although our evaluation primarily relies on the CICAPT-IIoT dataset, this dataset offers a highly representative testbed for real-world industrial environments. It integrates both physical controllers (e.g., Raspberry Pi and PLC simulators) and virtualized components running standard industrial protocols such as MODBUS and MQTT, forming a hybrid IT/OT setting.

Moreover, its attack scenarios are derived from APT29 campaigns and span over twenty TTPs across collection, exfiltration, and lateral movement phases, providing a balanced distribution of temporal and structural patterns. This diversity enables a realistic validation of the proposed cross-modal framework, as it captures both industrial and enterprise-level behavioral variance within a single integrated dataset.

Nevertheless, to rigorously establish the framework’s external generalization, future work will extend validation to datasets with different operational and threat characteristics. We plan to assess performance on SC-APT-2023 and the large-scale DARPA OpTC dataset [28], which represent enterprise-grade provenance traces and high-volume streaming conditions. Such evaluations will confirm the model’s adaptability beyond the ICS/OT context and quantitatively measure its resilience to domain shift and unseen attack patterns.

Furthermore, the choice of a 15-minute fixed-length window, while effective, could be enhanced. Our model is trained to detect malicious fragments within these windows, which can then be correlated by higher-level systems. However, future research should explore adaptive or overlapping windowing strategies to more robustly capture stealthy attacks of varying durations.

By providing a more holistic and context-aware analysis of system provenance data, this approach represents a significant step forward in protecting critical infrastructure against sophisticated, stealthy cyber threats.

Acknowledgment

This work was supported by the Nuclear Safety Research Program through Korea Foundation of Nuclear Safety (KoFONS) using the financial resource granted by the Nuclear Safety and Security Commission (NSSC) of the Republic of Korea (Grant number: 2106061, 50%). and This research was supported by the Institute of Information & Communications Technology Planning & Evaluation (IITP) grant funded by the Korea government(MSIT) (RS-2025-25394739, Development of Security Enhancement Technology for Industrial Control Systems Based on S/HBOM Supply Chain Protection, 50%).

References

[1] Dark Reading. 2024. Attackers Breach IT-Based Networks Before Jumping to ICS/OT Systems. Dark Reading (November 2024). Retrieved June 2025 from https://www.darkreading.com/ics-ot-security/attackers-breach-network-provider-ot-ics-network.

[2] Dragos Inc. 2025. 2025 OT Cybersecurity Report. Dragos Inc., Hanover, MD. Retrieved June 2025 from https://www.dragos.com/ot-cybersecurity-year-in-review/.

[3] Dragos Inc. 2024. Protect Against the FrostyGoop ICS Malware Threat with OT Cybersecurity Basics. Dragos Blog (December 2024). Retrieved June 2025 from https://www.dragos.com/blog/protect-against-frostygoop-ics-malware-targeting-operational-technology/.

[4] Jason D. Christopher. 2024. 2024 State of ICS/OT Cybersecurity: Our Past and Our Future. SANS Institute Blog (October 2024). Retrieved June 2025 from https://www.sans.org/blog/the-2024-state-of-ics-ot-cybersecurity-our-past-and-our-future/

[5] Dragos Inc. 2020. Living off the Land in ICS/OT Cybersecurity. Dragos Blog (June 2020). Retrieved June 2025 from https://www.dragos.com/blog/industry-news/living-off-the-land-in-ics-ot-cybersecurity/

[6] Roberto Perdisci, Giorgio Giacinto, and Fabio Roli. 2014. Signature Based Intrusion Detection for Zero-Day Attacks: (Not) A Closed Chapter? In Proceedings of the 2014 IEEE International Conference on Communications (ICC '14). IEEE, London, UK, 6759203. DOI: https://doi.org/10.1109/ICC.2014.6759203

[7] Dragos Inc. 2018. TRISIS Malware: Analysis of Safety System Targeted Malware. Technical Report. Dragos, Inc.

[8] FireEye. 2017. TRITON Actor TTP Profile, Custom Attack Tools, Detections, and ATT&CK Mapping. FireEye Threat Intelligence.

[9] Du, M., Li, F., Zheng, G., & Srikumar, V. 2017. Deeplog: Anomaly detection and diagnosis from system logs through deep learning. In Proceedings of the 2017 ACM SIGSAC conference on computer and communications security (pp. 1285-1298).

[10] Van Ede, T., Aghakhani, H., Spahn, N., Bortolameotti, R., Cova, M., Continella, A., ... & Vigna, G. (2022, May). Deepcase: Semi-supervised contextual analysis of security events. In 2022 IEEE Symposium on Security and Privacy (SP) (pp. 522-539). IEEE.

[11] Ongun, T., Stokes, J. W., Or, J. B., Tian, K., Tajaddodianfar, F., Neil, J., ... & Platt, J. C. (2021, October). Living-off-the-land command detection using active learning. In Proceedings of the 24th International Symposium on Research in Attacks, Intrusions and Defenses (pp. 442-455).

[12] Barr-Smith, F., Ugarte-Pedrero, X., Graziano, M., Spolaor, R., & Martinovic, I. (2021, May). Survivalism: Systematic analysis of windows malware living-off-the-land. In 2021 IEEE Symposium on Security and Privacy (SP) (pp. 1557-1574). IEEE.

[13] Alsaheel, A., Nan, Y., Ma, S., Yu, L., Walkup, G., Celik, Z. B., ... & Xu, D. (2021). {ATLAS}: A sequence-based learning approach for attack investigation. In 30th USENIX security symposium (USENIX security 21) (pp. 3005-3022).

[14] Villarreal-Vasquez, Miguel, et al. "Hunting for insider threats using LSTM-based anomaly detection." IEEE Transactions on Dependable and Secure Computing 20.1: 451-462. (2021)

[15] Li, Linrui, and Wen Chen. "ConGraph: Advanced persistent threat detection method based on provenance graph combined with process context in cyber-physical system environment." Electronics 13.5 (2024): 945.

[16] Wang, Q., Hassan, W. U., Li, D., Jee, K., Yu, X., Zou, K., ... & Chen, H. (2020, February). You Are What You Do: Hunting Stealthy Malware via Data Provenance Analysis. In NDSS.

[17] Han, X., Pasquier, T., Bates, A., Mickens, J., & Seltzer, M. "Unicorn: Runtime provenance-based detector for advanced persistent threats." arXiv preprint arXiv:2001.01525 (2020).

[18] Han, X., Yu, X., Pasquier, T., Li, D., Rhee, J., Mickens, J., ... & Chen, H. (2021). {SIGL}: Securing software installations through deep graph learning. In 30th USENIX Security Symposium (USENIX Security 21) (pp. 2345-2362).

[19] Zengy, J., Wang, X., Liu, J., Chen, Y., Liang, Z., Chua, T. S., & Chua, Z. L. (2022, May). Shadewatcher: Recommendation-guided cyber threat analysis using system audit records. In 2022 IEEE symposium on security and privacy (SP) (pp. 489-506). IEEE.

[20] Wang, S., Wang, Z., Zhou, T., Sun, H., Yin, X., Han, D., ... & Yang, J. "Threatrace: Detecting and tracing host-based threats in node level through provenance graph learning." IEEE Transactions on Information Forensics and Security 17 (2022): 3972-3987.

[21] Manzoor, E., Milajerdi, S. M., & Akoglu, L. (2016, August). Fast memory-efficient anomaly detection in streaming heterogeneous graphs. In Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining (pp. 1035-1044).

[22] Yang, D., Li, B., Rettig, L., & Cudré-Mauroux, P. (2017, November). Histosketch: Fast similarity-preserving sketching of streaming histograms with concept drift. In 2017 IEEE International Conference on Data Mining (ICDM) (pp. 545-554). IEEE.

[23] Yang, F., Xu, J., Xiong, C., Li, Z., & Zhang, K. (2023). {PROGRAPHER}: An anomaly detection system based on provenance graph embedding. In 32nd USENIX Security Symposium (USENIX Security 23) (pp. 4355-4372).

[24] Gysel, P., Wüest, C., Nwafor, K., Jašek, O., Ustyuzhanin, A., & Divakaran, D. M. EagleEye: Attention to Unveil Malicious Event Sequences from Provenance Graphs. arXiv preprint arXiv:2408.09217. (2024)

[25] Cheng, W., Zhu, T., Xiong, C., Sun, H., Wang, Z., Jing, S., ... & Chen, Y. SoK: Knowledge is All You Need: Accelerating Last Mile Delivery for Automated Provenance-based Intrusion Detection with LLMs. arXiv preprint arXiv:2503.03108. (2025)

[26] Huang, Y., Hassan, W. U., Guo, Y., Chen, X., & Li, D. PROVSYN: Synthesizing Provenance Graphs for Data Augmentation in Intrusion Detection Systems. arXiv preprint arXiv:2506.06226. (2025).

[27] Ghiasvand, E., Ray, S., Iqbal, S., Dadkhah, S., & Ghorbani, A. A. CICAPT-IIOT: A provenance-based APT attack dataset for IIoT environment. arXiv preprint arXiv:2407.11278. (2024)

[28] M. van Opstal and W. Arbaugh, "Operationally Transparent Cyber (OpTC) Data Release," 2019, https://github.com/FiveDirections/OpTC-data