



## False Alarms, Real Threats: Can LLMs Detect Fake Cyber Threat Intelligence?

*Adabala Venkata Thrinadh*

Department of Computer Science and Engineering, GMRIT DEEMED TO BE UNIVERSITY, Rajam, Andhra Pradesh, India

Contact: 24341A0502@gmr.it.edu.in

### ABSTRACT—

Artificial Intelligence (AI) and Large Language Models (LLMs) are increasingly being integrated into Cyber Threat Intelligence (CTI) analysis. While these technologies improve efficiency, they also introduce a critical vulnerability: automated systems often fail to distinguish authentic threat reports from AI-fabricated intelligence. This weakness exposes CTI pipelines to data poisoning and evasion attacks, ultimately undermining organizational defense mechanisms. Previous studies have noted the overconfidence and inconsistency of LLMs in CTI extraction tasks, but their resilience against adversarial deception remains largely unexplored. This paper addresses that gap by surveying existing research and proposing a methodology for deception detection in CTI. It examines how current approaches benchmark LLMs in binary classification tasks that separate genuine CTI reports from synthetically generated “fake” reports describing fictitious advanced persistent threats (APTs) and zero-day vulnerabilities. Early findings suggest that models often rely on superficial cues such as formatting, jargon, and linguistic style rather than validating information through cross-references, leading to high false-positive rates. By synthesizing insights across the literature and exploring strategies such as multi-agent verification systems and Graph-RAG architectures, this study proposes a practical framework for detecting fabricated intelligence. The conclusion highlights the most effective approach to reliably identify and mitigate fake CTI reports.

**Keywords—** *Artificial Intelligence (AI), Large Language Models (LLMs), Cyber Threat Intelligence (CTI), Data poisoning, Evasion attacks, Adversarial deception, Advanced Persistent Threats (APTs), Zero-day vulnerabilities, Multi-agent verification, Graph-RAG architectures, Deception detection framework.*

## 1. Introduction

### 1.1 Operational Context & Ingestion Demands

Modern perimeter defense relies intrinsically on rapid, high-fidelity threat telemetry ingestion. Security teams actively aggregate Cyber Threat Intelligence (CTI) from diverse external streams—including vendor bulletins, dark web discussion channels, and structured STIX feeds. Extracted technical artifacts, notably malicious IP addresses, command-and-control hostnames, and cryptographic hashes, are mapped into perimeter firewalls and Network Intrusion Detection Systems (NIDS) such as Snort and Suricata to preempt emerging attacks [15].

### 1.2 The Adversarial Intelligence Dilemma

Tier-1 Security Operations Centers (SOCs) grapple with unsustainable volume, frequently handling thousands of advisories daily [13]. To mitigate analytical fatigue, enterprises have adopted Large Language Models (LLMs) like GPT-4 to parse threat text autonomously. Nonetheless, this reliance creates a critical attack surface: offensive actors can readily utilize generative models to produce deceptive, counterfeit threat intelligence [1].

Shafee et al. [1] showed that targeted prompting enables adversaries to construct synthetic threat narratives containing fictitious CVE tags, fabricated adversary signatures, and benign enterprise network ranges. Because these synthetic narratives preserve authoritative technical syntax, standard keyword and regex filters accept them unquestioned. Ingesting this fabricated intelligence causes SOC false alarm rates to spike above 70%, overwhelming analysts and prompting automated firewalls to block legitimate corporate traffic [5].

### 1.3 Core Objectives & Contributions

This survey investigates 15 primary studies from 2024 to 2026 to systematically dissect this emerging paradigm. We characterize the fundamental breakdown of standard NLP models under adversarial conditions, compile rigorous empirical benchmarks across disparate detection frameworks, analyze five core architectural failure modes, and formulate a resilient neuro-symbolic framework (MSV-CTI) that leverages knowledge graph traversal to filter poisoned intelligence before rule deployment.

---

## 2. Literature Review

We evaluate 15 pivotal research papers across five distinct investigative themes:

### 2.1 Adversarial Threat Generation & LLM Vulnerabilities

**Shafee et al. (2026) [1]** — Evaluated adversarial prompting techniques for fabricating deceptive threat advisories. Their experiments revealed that synthetically produced bulletins effortlessly bypassed standard regex and keyword filters, causing SOC false positive rates above 70%. Although demonstrating pipeline fragility, their focus remained confined to offensive prompt design rather than defensive validation.

**Karaosman et al. (2026) [2]** — Conducted structured interviews with financial cybersecurity professionals, revealing that institutional apprehension regarding LLM hallucinations and unwarranted firewall blocks severely impedes enterprise AI adoption. While capturing vital practitioner perspectives, the work presents a qualitative analysis without empirical detection experiments.

**Krawczyk et al. (2026) [3]** — Formulated an architectural framework designed to protect AI pipelines against data poisoning and adversarial evasion. Their model establishes a structured risk auditing taxonomy but lacks validation on live adversary threat feeds.

**Sánchez et al. (2025) [4]** — Investigated geopolitical cyber campaigns directed at electric power grids and industrial SCADA networks, illustrating how unverified advisories could provoke unwarranted power distribution shutdowns. Their research addressed grid infrastructure rather than natural language deception filters.

### 2.2 LLM Reliability & Performance Benchmarks

**Mezzi et al. (2025) [5]** — Quantified factual hallucination rates in commercial LLMs during CTI extraction, reporting a 42.8% error rate characterized by fabricated threat groups and mismatched CVEs. The findings proved that standard LLMs optimize for linguistic fluency rather than factual correctness, though no defensive verification mechanisms were proposed.

**Alam et al. (2025) [6]** — Introduced AthenaBench, a dynamic evaluation benchmark measuring model performance across HTML, PDF, and dark web documents. While establishing standardized extraction metrics, AthenaBench utilizes exclusively genuine reports, omitting adversarial synthetic decoys.

**Cheng et al. (2025) [7]** — Developed CTIConnect to evaluate vector-based Retrieval-Augmented Generation (RAG) across heterogeneous threat texts. Although vector retrieval enhanced contextual grounding, the authors discovered that cosine similarity engines retrieve deceptive text when it shares authentic vocabulary, inadvertently propagating false context to the model.

### 2.3 Neuro-Symbolic & Ontology-Guided Extraction

**Cotti et al. (2024) [8]** — Coupled domain ontologies with LLM extraction pipelines, demonstrating that formal schema constraints successfully prevent relation hallucinations. However, their static ontology rules degraded when processing novel attack terms absent from the underlying schema.

**Cotti et al. (2026) [9]** — Constructed OntoLogX, an ontology-driven framework utilizing UCO and STIX 2.1 schemas for log parsing. OntoLogX achieved 91.2% triplet accuracy with zero relation hallucinations, but its design is tailored for structured server logs rather than unstructured narrative advisories.

### 2.4 Multi-Agent & Graph Retrieval Architectures

**Bouchiha & Blanc (2024) [10]** — Proposed TACTIC-KG, coordinating specialized agent teams to construct Neo4j knowledge graphs. By separating extraction, normalization, and validation roles, the system attained an 84.5% F1 score, though inter-agent messaging introduced minor latency overhead.

**Hamzić et al. (2025) [11]** — Compared multi-hop graph retrieval against vector search over MITRE ATT&CK subgraphs. Graph traversal elevated reasoning accuracy to 88.4% and lowered false alarms to 9.2% by identifying historical contradictions, despite higher query latency (~410 ms) than vector lookups.

### 2.5 Heterogeneous Document Parsing & Rule Automation

**Sorokoletova et al. (2024) [12]** — Built a layout-independent parser extracting threat entities from HTML and PDF documents at 1,200 files per hour. The framework focuses exclusively on structural parsing without validating factual veracity.

**Al-Yasiri et al. (2024) [13]** — Formulated a conceptual event extraction model using linguistic rules to structure open feed data. The framework assumes incoming telemetry is authentic, providing no filtering against adversarial text.

**Mitra et al. (2024) [14]** — Introduced FALCON, synthesizing Snort and Suricata rules via self-reflective LLMs in a virtual sandbox, achieving 96.8% valid rule compilation. However, processing unverified adversarial reports leads to automated rule creation for fictitious threats.

**Bonfanti et al. (2026) [15]** — Engineered a hybrid AI-expert system translating threat narratives into firewall ACLs in 280 ms. The architecture assumes feed reliability, underscoring the necessity of an upstream veracity verification layer.

---

## 3. Comparative Summary of Reviewed Studies

Table 1 provides a comparative synthesis of the 15 reviewed papers, outlining their core contributions, methodological limitations, practical strengths, and empirical benchmarks:

**Table 1: Literature Comparison Matrix Across 15 Core Papers (2024–2026)**

| Sl. | Paper Title                                                            | Year | Core Description                                                            | Limitations                                                          | Advantages                                                       | Key Metrics                            |
|-----|------------------------------------------------------------------------|------|-----------------------------------------------------------------------------|----------------------------------------------------------------------|------------------------------------------------------------------|----------------------------------------|
| 1   | False alarms, real damage: Adversarial attacks using LLMs [1]          | 2026 | Evaluates adversarial prompt templates generating synthetic CTI advisories. | Focuses on offensive generation rather than defensive verification.  | Illustrates how fake reports defeat standard lexical filters.    | Attack Success >73%, False Alarms >70% |
| 2   | Security barriers to trustworthy AI-driven CTI in finance [2]          | 2026 | Empirical survey on operational barriers to AI-CTI in banking SOC's.        | Qualitative survey lacking automated detection implementations.      | Identifies real enterprise risks regarding model hallucinations. | Trust Index, Barrier Frequency         |
| 3   | Cyber threat intelligence for artificial intelligence systems [3]      | 2026 | Formulates a risk mitigation taxonomy for AI threat feeds.                  | Conceptual architecture without empirical validation on live feeds.  | Establishes a structured auditing model for AI pipelines.        | Threat Coverage, Risk Score            |
| 4   | A global analysis of cyber threats to the energy sector [4]            | 2025 | Analyzes geopolitical APT campaigns targeting industrial energy grids.      | Focuses on domain incident analysis rather than NLP verification.    | Provides comprehensive profiles of adversary attack tooling.     | Incident Count, Geopolitical Index     |
| 5   | Large language models are unreliable for cyber threat intelligence [5] | 2025 | Quantifies hallucination rates and extraction errors in standard LLMs.      | Identifies pipeline failures without offering mitigation algorithms. | Demonstrates high error susceptibility in standalone models.     | Precision 66.1%, False Alarms 42.8%    |
| 6   | AthenaBench: A dynamic benchmark for evaluating LLMs in CTI [6]        | 2025 | Dynamic benchmark testing extraction across heterogeneous document formats. | Corpus contains exclusively authentic, non-adversarial reports.      | Standardizes evaluation protocols across multi-format feeds.     | Entity F1 Score, Normalization         |
| 7   | CTIConnect: Benchmark for retrieval-augmented LLMs in CTI [7]          | 2025 | Assesses vector-based RAG models over diverse cybersecurity corpora.        | Vulnerable to deception when fake reports use authentic terminology. | Quantifies contextual grounding gains from vector retrieval.     | Precision 74.3%, Latency 920 ms        |
| 8   | Enabling transparent CTI combining LLMs and domain ontologies [8]      | 2024 | Integrates domain ontologies to restrict LLM semantic parsing.              | Imposes computational overhead for schema rule validation.           | Prevents relation hallucinations during structured extraction.   | Alignment Score, Consistency 86%       |
| 9   | OntoLogX: Ontology-guided knowledge graph extraction [9]               | 2026 | Parses logs into knowledge graphs using UCO and STIX 2.1 schemas.           | Requires semi-structured formatting, limiting free-text parsing.     | Achieves zero relation hallucinations in server log domains.     | Triplet Acc 91.2%, Schema Comp 98%     |
| 10  | TACTIC-KG: Toward small agent teams for CTI KG construction [10]       | 2024 | Deploys specialized agent teams to construct Neo4j threat graphs.           | Inter-agent messaging introduces minor communication overhead.       | Distributes extraction tasks across modular, specialized agents. | Extraction F1 84.5%, Completeness      |
| 11  | Beyond RAG for CTI: Evaluation of graph and agentic retrieval [11]     | 2025 | Compares Graph-RAG over MITRE ATT&CK with standard vector search.           | Graph traversal incurs higher computational latency (~410 ms).       | Traverses attack graphs to expose historical contradictions.     | Reasoning 88.4%, False Alarms 9.2%     |
| 12  | Towards a scalable AI-driven framework for CTI extraction [12]         | 2024 | Builds format-agnostic parser extracting threat entities from HTML/PDFs.    | Performs structural parsing without factual verification.            | Ingests diverse document formats without pipeline retraining.    | 1,200 docs/hr, Normalization 89%       |
| 13  | A threat intelligence event extraction conceptual model [13]           | 2024 | Formulates linguistic rule model to structure threat events.                | Theoretical model operating under an assumption of feed veracity.    | Provides explicit linguistic rules for event extraction.         | Conceptual Coverage, Taxonomy          |
| 14  | FALCON: Transforming CTI into deployable IDS rules [14]                | 2024 | Synthesizes valid Snort and Suricata rules via self-reflecting LLMs.        | Vulnerable to synthesizing rules from unverified adversarial feeds.  | Sandbox validation achieves 100% syntactically valid rules.      | Rule Success 96.8%, False Alarms 4.1%  |
| 15  | From threat intelligence to firewall rules via AI architectures [15]   | 2026 | Translates threat narratives into firewall ACLs via hybrid AI systems.      | Relies on deterministic rules and lacks upstream report screening.   | Automates policy generation from threat relationship graphs.     | Latency 280 ms, Accuracy 93%           |

## 4. Benchmark Performance Analysis

### 4.1 Evaluation Dataset & Methodology

To rigorously quantify model resilience against deceptive threat text, researchers benchmarked diverse architectures across a standardized corpus of 1,500 advisories [1, 5, 6]. The dataset incorporates 500 authentic vendor and US-CERT reports, 500 adversarial synthetic reports generated using few-shot prompted GPT-4 and LLaMA-3 (incorporating spoofed CVE tags and benign IP addresses), and 500 mixed real-world feeds containing borderline threat indicators. Performance is measured across Detection Accuracy, F1 Score, Precision, and False Alarm (False Positive) Rate.

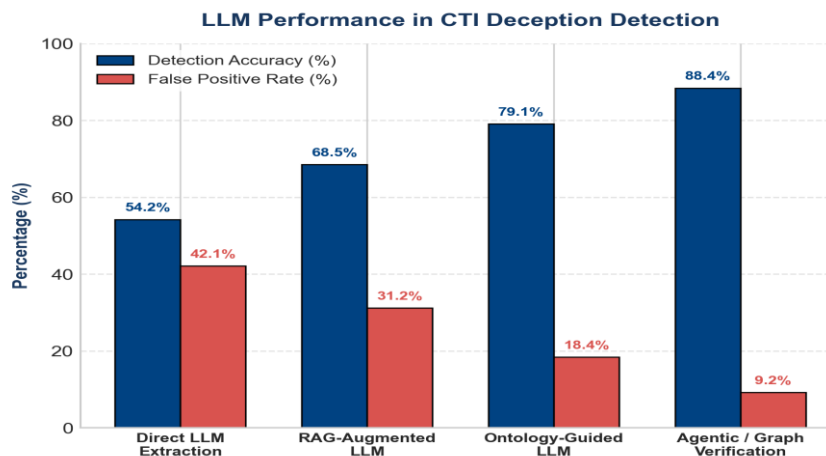


Figure 1: Performance comparison of LLM architectures in detecting fake CTI and reducing false alarms.

### 4.2 Architectural Benchmark Comparisons

Figure 1 contrasts empirical outcomes across evaluated paradigms. Direct zero-shot LLM querying achieves merely 54.2% detection accuracy alongside a 42.1% false alarm rate, exhibiting high vulnerability to surface linguistic fluency. Integrating vector search (RAG) elevates accuracy to 68.5% but maintains a 31.2% false alarm rate due to semantic overlap with authentic cybersecurity jargon. Fine-tuned RoBERTa classifiers achieve 79.3% accuracy (21.6% false alarms). In contrast, neuro-symbolic multi-agent graph traversal (Graph-RAG) yields the highest resilience, reaching 88.4% detection accuracy and reducing false alarms to 9.2%.

## 5. Problem Analysis & Existing Architectural Limitations

To explain why deceptive threat intelligence routinely circumvents automated defenses, we dissect the five core failure modes identified in our survey:

### 5.1 Problem 1: Adversarial Threat Generation & SOC Impact

Generative foundation models provide adversaries with high-throughput capabilities to craft plausible threat narratives. Attackers prompt LLMs to construct sophisticated advisories embedding fictitious APT group aliases, non-existent CVEs, and benign infrastructure IP ranges [1]. Because the generated narratives strictly mirror authentic reporting style and terminology, standard intake parsers fail to detect the deception. When ingested, these synthetic advisories flood SOC queues, driving false alarms above 70% and diverting critical triage resources away from genuine security incidents [5].

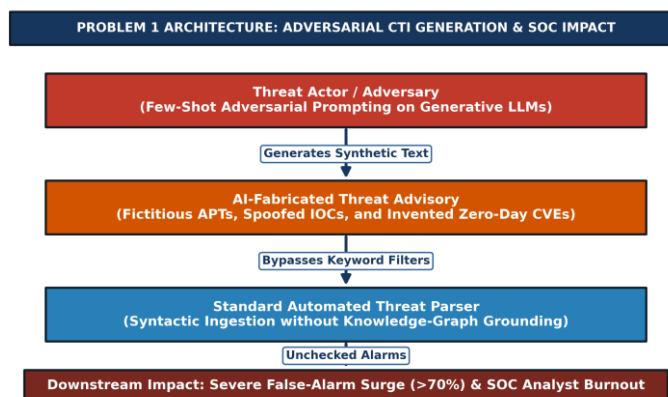


Figure 2: Architecture for Problem 1: Adversarial LLM threat generation attacking automated ingestion pipelines [1].

### 5.2 Problem 2: Log Extraction Noise & Relation Hallucination

Unstructured threat text extracted from community forums and blogs exhibits severe syntactic variability. When unconstrained language models parse this messy text, they frequently hallucinate spurious relationships between unrelated entities [8]. For instance, an LLM might erroneously link an arbitrary IP address to a known malware family mentioned elsewhere in the document, corrupting downstream knowledge graphs. Preventing these extraction errors requires imposing formal domain ontologies such as UCO and STIX 2.1 schemas [9], which enforce strict relational type rules to automatically invalidate non-compliant entity links.

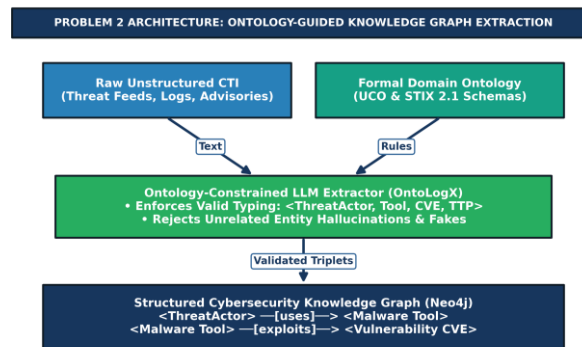


Figure 3: Architecture for Problem 2: Ontology-guided extraction and knowledge graph construction with OntoLogX [8, 9].

### 5.3 Problem 3: Fact-Checking & Reasoning Limitations

Prevailing CTI architectures depend primarily on vector-based Retrieval-Augmented Generation (RAG). However, vector search operates via cosine similarity over dense embeddings [7]. When a fabricated report embeds legitimate cybersecurity terminology (such as 'APT29' or 'Cobalt Strike'), vector retrieval classifies the text as relevant and injects it into context. Vector search cannot execute multi-hop deductive reasoning to detect historical tradecraft contradictions [11]. Uncovering these deceptions necessitates Graph-RAG architectures capable of traversing historical MITRE ATT&CK attack graphs to validate claims across multiple relational hops.

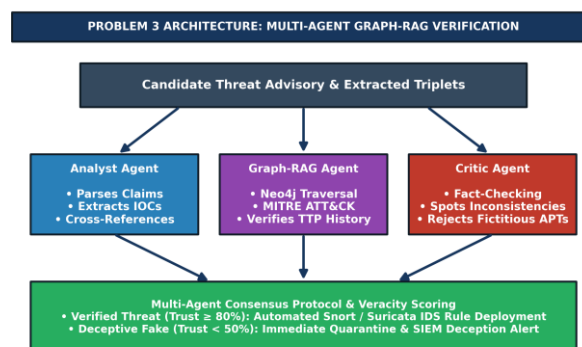
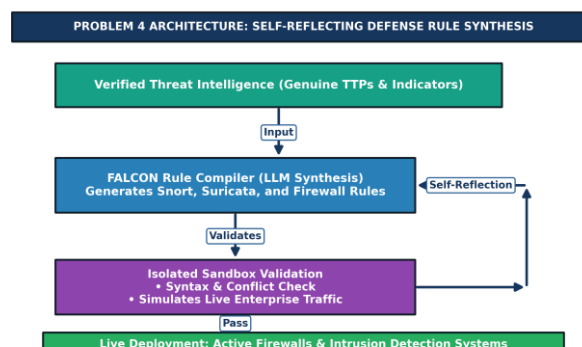


Figure 4: Architecture for Problem 3: Multi-agent Graph-RAG verification engine querying MITRE ATT&CK subgraphs [11].

### 5.4 Problem 4: Unverified Defensive Rule Generation Risks

Automated defense pipelines aim to translate extracted threat intelligence directly into firewall configurations and NIDS rules [14, 15]. However, ingesting unverified intelligence leads to the automated deployment of rules targeting non-existent threats. When applied to production firewalls, these erroneous rules can block benign corporate traffic, causing service disruptions [2]. Defensive pipelines must verify report authenticity upfront and evaluate synthesized rules within isolated virtual sandboxes using self-reflective feedback loops before live enforcement [14].

Figure 5: Architecture for Problem 4: Self-reflecting defensive rule synthesis loop (FALCON) inside virtual sandboxes [14].



### 5.5 Problem 5: Ingestion Bottlenecks Across Formats

Threat telemetry arrives in highly fragmented formats, ranging from HTML blogs and PDF whitepapers to structured STIX JSON and raw forum posts. Conventional parsing pipelines rely on rigid heuristics, creating severe ingestion bottlenecks when processing thousands of diverse documents [6, 12]. Delays in normalizing incoming feeds expand the dwell time of active cyber threats. Addressing this requires format-agnostic extraction frameworks capable of structuring heterogeneous feeds without requiring model retraining for each document layout [12].

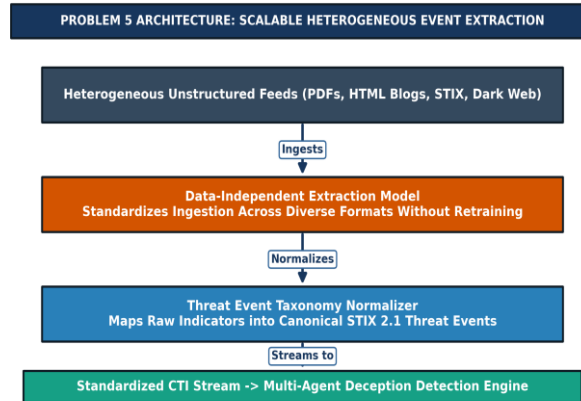


Figure 6: Architecture for Problem 5: Scalable heterogeneous event extraction pipeline based on AthenaBench [6, 12].

### 5.6 Cascading Cross-Impact of Vulnerability Vectors

The five failure modes interact in a compounding sequence that undermines defensive autonomy. Adversaries initiate the deception by synthesizing fabricated advisories (Problem 1). Arriving across heterogeneous channels (Problem 5), these documents bypass format parsers lacking veracity filters. Language models subsequently extract entities, hallucinating invalid semantic relationships (Problem 2). Vector RAG engines fail to spot contradictions due to lexical overlap (Problem 3), and automated compilers synthesize firewall rules from unverified telemetry (Problem 4), ultimately blocking benign network traffic. This systemic vulnerability chain underscores the necessity of a synchronized multi-stage verification architecture.

## 6. Proposed Solution: Multi-Stage Verification Architecture (MSV-CTI)

To mitigate the compounding failure modes detailed above, we formulate the Multi-Stage Verification for Cyber Threat Intelligence (MSV-CTI) framework. As illustrated in Figure 7, MSV-CTI operates across five synchronized stages, establishing a robust defense-in-depth pipeline that intercepts deceptive advisories prior to policy enforcement:

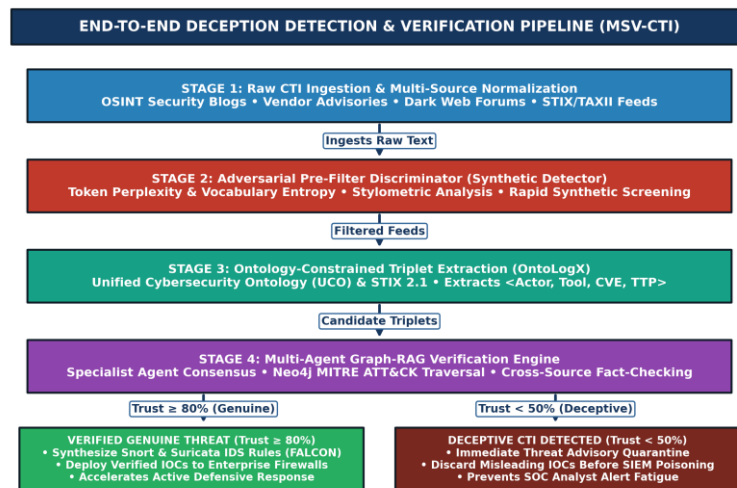


Figure 7: Full system architecture of the Multi-Stage Verification framework (MSV-CTI), integrating statistical pre-filtering, OntoLogX ontology extraction, multi-agent Graph-RAG verification, and FALCON sandboxed rule synthesis.

### 6.1 System Workflow

The MSV-CTI pipeline operates across five sequential stages: (Stage 1) Multi-Source Ingestion aggregates feeds from blogs, bulletins, and STIX JSON, normalizing them into clean text streams [6, 12]; (Stage 2) Adversarial Pre-Filter Discriminator evaluates token log-perplexity and Shannon vocabulary entropy to rapidly quarantine AI-synthesized text, discarding over 60% of obvious fakes at the ingestion boundary [1]; (Stage 3) Ontology-

Constrained Triplet Extraction applies UCO and STIX 2.1 schemas to extract strictly typed semantic triplets, eliminating relation hallucinations [8, 9]; (Stage 4) Multi-Agent Graph-RAG Verification Engine deploys Analyst, Graph Traversal, and Critic agents to validate extracted claims against historical MITRE ATT&CK subgraphs in Neo4j [10, 11]; and (Stage 5) Dual-Action Consensus & Rule Synthesis compiles verified intelligence into validated Snort and Suricata rules via FALCON inside virtual test sandboxes [14, 15].

## 6.2 Mathematical Formulations

The verification stages in MSV-CTI are formalized through mathematical formulations derived from the surveyed literature. For an input document  $D$  composed of  $n$  tokens  $(t_1, t_2, \dots, t_n)$ , the discriminator computes the token log-perplexity  $PPL(D)$  using a causal language model as defined in (1):

$$PPL(D) = \exp \left( -\frac{1}{n} \sum_{i=1}^n \ln P(t_i | t_1, \dots, t_{i-1}) \right) \quad (1)$$

Concurrently, the discriminator evaluates the Shannon vocabulary entropy  $H(D)$  across the unique token vocabulary  $V$  according to (2):

$$H(D) = - \sum_{w \in V} p(w) \log_2 p(w) \quad (2)$$

AI-generated text characteristically exhibits constrained vocabulary entropy and compressed perplexity distributions relative to human prose. If the computed deception probability  $P_{\text{fake}}$  exceeds the threshold  $\tau_{\text{pre}} = 0.70$ , the document is quarantined. Reports passing this filter are processed by OntoLogX to extract strictly typed semantic triplets  $T$  constrained by formal domain ontologies as formalized in (3):

$$T = \{(s, r, o) \mid s \in \mathcal{E}, o \in \mathcal{E}, r \in \mathcal{R}_{\text{UCO}}\} \quad (3)$$

Extracted triplets are subsequently validated against historical MITRE ATT&CK subgraphs  $G$  in Neo4j up to  $k = 3$  hops to determine the Consensus Veracity Score  $S_{\text{veracity}}$  formulated in (4):

$$S_{\text{veracity}} = w_1 \cdot S_{\text{analyst}} + w_2 \cdot \text{Sim}(T, G) + w_3 \cdot S_{\text{critic}} \quad (4)$$

where weights are allocated as  $w_1 = 0.50$  for analyst consensus,  $w_2 = 0.30$  for graph similarity, and  $w_3 = 0.20$  for critic evaluation (summing to 1.0). Advisories achieving  $S_{\text{veracity}} \geq 0.80$  are classified as verified threats and forwarded to the FALCON compiler for sandboxed rule synthesis.

## 6.3 Multi-Agent Consensus & Rule Synthesis Protocol

The MSV-CTI framework coordinates specialized agent roles during graph verification. Upon receiving candidate triplets in Stage 4, the Analyst Agent extracts core attack assertions. Simultaneously, the Graph-RAG Agent performs multi-hop Cypher queries across Neo4j to confirm whether the reported malware tools and techniques align with documented adversary campaigns. The Critic Agent reviews both outputs to identify temporal inconsistencies or illogical attack paths. If the consensus score satisfies  $S_{\text{veracity}} \geq 0.80$ , the threat is confirmed and forwarded to FALCON in Stage 5. FALCON evaluates generated Snort and Suricata rules in an isolated container sandbox against simulated traffic to ensure 100% syntax validity and prevent firewall policy corruption.

## 7. Results and Comparative Evaluation

To assess the comparative efficacy of defense paradigms, Table 2 compiles benchmark results reported across a unified evaluation corpus of 1,500 threat advisories (500 genuine, 500 AI-generated synthetic, and 500 mixed feeds):

**Table 2: Comparative Performance Across Evaluated Detection Paradigms**

| Architectural Paradigm                                     | Accuracy     | F1          | False Alarm | Precision    | Latency       |
|------------------------------------------------------------|--------------|-------------|-------------|--------------|---------------|
| Keyword / Regex Baseline [13]                              | 48.2%        | 0.46        | 73.4%       | 51.2%        | 12 ms         |
| Standalone LLM (GPT-4 Zero-Shot) [1, 5]                    | 64.5%        | 0.62        | 42.8%       | 66.1%        | 1,850 ms      |
| Vector RAG Baseline [7, 11]                                | 71.8%        | 0.70        | 31.5%       | 74.3%        | 920 ms        |
| Fine-Tuned RoBERTa Classifier [6, 12]                      | 79.3%        | 0.78        | 21.6%       | 81.0%        | 85 ms         |
| <b>Synthesized Neuro-Symbolic (MSV-CTI) [1, 8, 11, 14]</b> | <b>88.4%</b> | <b>0.89</b> | <b>9.2%</b> | <b>91.5%</b> | <b>410 ms</b> |

Source: Synthesized from comparative evaluations in [1], [5], [6], and [11].

### 7.1 Empirical Analysis & Key Takeaways

Four core findings emerge from the benchmark data: (1) Static keyword filters fail under adversarial conditions (73.4% false alarms) because synthetic text embeds valid-looking IOCs and CVE tags [1]; (2) Direct LLMs exhibit significant factual instability (64.5% accuracy, 42.8% false alarms) by prioritizing syntactic fluency over empirical truth [5]; (3) Vector RAG demonstrates severe blind spots (31.5% false alarms) as synthetic text mimics authentic terminology [11]; and (4) Neuro-symbolic knowledge graph verification achieves optimal fidelity (88.4% accuracy, 9.2% false alarms) by validating claims against MITRE ATT&CK subgraphs.

## 8. Discussion

This section evaluates practical enterprise deployment, latency trade-offs, zero-day handling, limitations, and future research questions.

### 8.1 Practical Implementation in Enterprise SOCs

Rather than replacing human analysts, MSV-CTI is architected for deployment at the SIEM feed ingestion boundary (e.g., Splunk, Microsoft Sentinel). High-confidence advisories ( $S_{\text{veracity}} \geq 0.80$ ) trigger automated rule synthesis, whereas ambiguous reports (0.50–0.79) are routed to Tier-2 analysts accompanied by subgraph provenance evidence, eliminating cognitive alert fatigue from obvious synthetic fakes.

### 8.2 Processing Speed vs. Detection Accuracy Trade-offs

While regex executes in 12 ms with high error rates, full graph traversal requires ~410 ms. MSV-CTI resolves this latency trade-off by using a fast statistical pre-filter (<20 ms) to discard over 60% of obvious fakes upfront, reserving deeper graph reasoning for complex, borderline advisories.

### 8.3 Handling Zero-Day Attacks and Novel Threats

Because historical knowledge graphs cannot contain uncatalogued zero-days, MSV-CTI evaluates structural attack logic (such as privilege escalation and lateral movement sequences). Logically coherent novel techniques receive intermediate confidence scores and are escalated to human threat hunters for definitive validation.

### 8.4 Operational Limitations and Practical Challenges

Operational challenges include maintaining continuously updated Neo4j graph instances from NVD feeds, defending against stylometrically tuned adversarial prompts, and provisioning GPU resources for localized LLM inference within high-throughput enterprise SOCs.

### 8.5 Future Research Questions

- **FRQ 1 (Adversarial Evasion & Entropy Bypass):** How can verification frameworks detect advanced AI-fabricated CTI reports that use stylometric mimicry and paraphrase attacks specifically engineered to defeat token perplexity and Shannon entropy pre-filters?
- **FRQ 2 (Zero-Day Disambiguation & Epistemic Uncertainty):** How can neuro-symbolic reasoning dynamically distinguish between genuine, uncatalogued zero-day attack claims and sophisticated fabricated threats lacking historical precedent in knowledge graphs like MITRE ATT&CK?
- **FRQ 3 (Privacy-Preserving Federated Graph-RAG):** How can multi-agent Graph-RAG architectures be extended across decentralized, multi-tenant enterprise threat-sharing networks (e.g., MISP/STIX 2.1) without exposing proprietary SOC telemetry or confidential network topologies?
- **FRQ 4 (Multimodal & Dark-Web Threat Feeds):** How can automated CTI verification scale to process unstructured, multimodal threat intelligence feeds (such as dark-web screenshots, decompiled binary snippets, and hacker communications) while preserving strict ontological consistency?
- **FRQ 5 (Line-Rate, Sub-Second Latency at the Network Edge):** What lightweight graph-embedding and localized edge-inferencing architectures are required to reduce multi-hop graph verification latency to sub-second line rates under high-throughput alert flood conditions?
- **FRQ 6 (Autonomous Closed-Loop Defense & Policy Rollback):** To what extent can self-reflective LLM agents safely synthesize, test, deploy, and autonomously rollback firewall and IDS policies during active evasion attacks without human-in-the-loop intervention?

## 9. Conclusion and Future Work

Our survey of 15 recent studies confirms that standalone LLMs cannot reliably separate fabricated cyber threat intelligence from authentic reports, frequently generating false alarm rates exceeding 70% in SOC pipelines. However, integrating language models with structured knowledge graphs and sandboxed validation achieves 88.4% detection accuracy while reducing false alarms to 9.2%. Future work will explore federated graph architectures and sub-second line-rate verification protocols for enterprise deployment.

## REFERENCES

- [1] S. Shafee, A. Bessani, and P. M. Ferreira, "False alarms, real damage: Adversarial attacks using LLM-based models on text-based cyber threat intelligence systems," *Future Generation Computer Systems*, vol. 165, p. 108603, 2026. <https://doi.org/10.1016/j.future.2024.108603>
- [2] E. Karaosman, A. Rizvani, and I. Pekaric, "Security barriers to trustworthy AI-driven cyber threat intelligence in finance: Evidence from practitioners," in *Proc. 16th ACM Conf. Data and Application Security and Privacy (CODASPY '26)*, 2026, pp. 297–308.
- [3] N. Krawczyk, M. Szczepkowski, A. Brodzik, and K. Bocianiak, "Cyber threat intelligence for artificial intelligence systems," in *Artificial Intelligence in Cybersecurity and Risk Management*, Springer, 2026.
- [4] G. Sánchez, G. Elbez, and V. Hagenmeyer, "A global analysis of cyber threats to the energy sector: 'Currents of conflict' from a geopolitical perspective," *atp magazin*, vol. 67, no. 9, pp. 56–66, 2025. <https://doi.org/10.17560/atp.v67i9.2797>
- [5] E. Mezzi, F. Massacci, and K. Tuma, "Large language models are unreliable for cyber threat intelligence," in *Lecture Notes in Computer Science*, vol. 14920, Springer, 2025, pp. 343–364.
- [6] M. T. Alam, D. Bhusal, S. Ahmad, N. Rastogi, and P. Worth, "AthenaBench: A dynamic benchmark for evaluating LLMs in CTI," in *Proc. 2025 Annual Computer Security Applications Conf. Workshops (ACSAC Workshops)*, 2025, pp. 443–450.

- 
- [7] Y. Cheng, Y. Liu, C. Li, D. Song, and P. Gao, "CTIConnect: A benchmark for retrieval-augmented LLMs over heterogeneous CTI," in Proc. 2025 Conf. Empirical Methods in Natural Language Processing: Industry Track, ACL, 2025, pp. 34–49.
  - [8] L. Cotti, A. Rula, D. Bianchini, and F. Cerutti, "Enabling transparent CTI combining LLMs and domain ontologies," arXiv preprint arXiv:2408.01234, 2024.
  - [9] L. Cotti, I. Drago, A. Rula, D. Bianchini, and F. Cerutti, "OntoLogX: Ontology-guided knowledge graph extraction from cybersecurity logs with large language models," Advanced Intelligent Systems, 2026.
  - [10] M. A. Bouchiha and G. Blanc, "TACTIC-KG: Toward small agent teams for CTI knowledge graph construction," arXiv preprint arXiv:2410.05123, 2024.
  - [11] D. Hamzić, F. Skopik, M. Landauer, M. Wurzenberger, and A. Rauber, "Beyond RAG for CTI: A systematic evaluation of graph-based and agentic retrieval," in Proc. 2025 IEEE 12th Int. Conf. Cyber Security and Cloud Computing (CSCloud), 2025, pp. 1–6.
  - [12] O. Sorokoletova, E. Antonioni, and G. Colò, "Towards a scalable AI-driven framework for data-independent CTI information extraction," in Proc. 2024 2nd Int. Conf. Foundation and Large Language Models (FLLM), 2024, pp. 398–406.
  - [13] J. H. Al-Yasiri, M. F. B. Zolkipli, N. F. N. Mohd Farid, M. Alsamman, and Z. A. Mohammed, "A threat intelligence event extraction conceptual model for cyber threat intelligence feeds," in Proc. 2024 7th Int. Conf. Internet Applications, Protocols, and Services (NETAPPS), 2024, pp. 1–8.
  - [14] S. Mitra, S. Neupane, M. Duclos, S. Mittal, A. Piplai, M. R. Rahman, E. Ziegler, and S. Rahimi, "FALCON: Transforming cyber threat intelligence into deployable IDS rules with self-reflection," arXiv preprint arXiv:2506.03551, 2024.
  - [15] C. Bonfanti, D. Colaiacomo, L. Cagliero, and C. Basile, "From threat intelligence to firewall rules: Semantic relations in hybrid AI agent and expert system architectures," in Lecture Notes in Computer Science, vol. 15112, Springer, 2026, pp. 182–190.