CIA Open Access

An Empirical Study of Large Language Models for Threat Intelligence Analysis and Incident Response

Ruoxi Jia[1], Jin Zhang[1.2] Julian Prescot[2]

[1] *Computer Science, Universtiy of Southern California, CA, USA*
[1.2] *Computer Science, Illinois Institute of Technology, IL, USA*
[2] *Computational Science, Princeton University, Princeton, NJ, USA*

DOI: 10.63575/CIA.2024.20109

Abstract

The exponential growth of cyber threats necessitates advanced automation in threat intelligence analysis and incident response workflows. This empirical study investigates the application of Large Language Models (LLMs) across critical security operations tasks, including threat intelligence extraction, TTP mapping, and automated response generation. Through systematic evaluation of multiple LLM architectures on real-world cybersecurity datasets comprising 1,000 threat intelligence reports and 500 incident records, we assess performance across entity extraction, threat actor attribution, and remediation recommendation tasks. Our experimental results demonstrate that LLMs achieve F1 scores exceeding 0.88 for Indicator of Compromise (IoC) extraction and reduce incident response time by 64% while maintaining 82% accuracy in MITRE ATT&CK technique mapping. The findings reveal significant efficiency gains with RAG-enhanced configurations showing 19% performance improvement over baseline approaches. This work provides empirical evidence supporting LLM deployment in security operations centers and identifies critical challenges in production environments.

Keywords: *Large Language Models, Threat Intelligence, Incident Response, Cybersecurity Automation*

1. Introduction

1.1. Research Background and Motivation

1.1.1. The Growing Challenge of Cyber Threat Intelligence Analysis

The contemporary threat landscape presents unprecedented challenges for cybersecurity professionals. Global cybercrime damages reached $8 trillion in 2023, with ransomware attacks increasing by 95% year-over-year[1]. Security Operations Centers (SOCs) process an average of 11,000 alerts daily, yet 67% remain uninvestigated due to resource constraints. The volume and sophistication of Advanced Persistent Threats (APTs) overwhelm traditional analysis capabilities, creating a critical gap between threat detection and effective response. Threat intelligence analysis requires synthesizing information from diverse sources including vulnerability databases, dark web forums, security advisories, and incident reports. The mean time to identify a breach stands at 197 days, while containment requires an additional 69 days, resulting in substantial financial and reputational damage[2].

1.1.2. Limitations of Traditional Manual Analysis Approaches

Conventional threat intelligence workflows rely heavily on manual effort and specialized expertise. Security analysts spend approximately 53% of their time on repetitive tasks such as alert triage, IoC validation, and report generation. The scarcity of skilled cybersecurity professionals, estimated at 3.4 million unfilled positions globally, exacerbates the challenge. Traditional rule-based systems and signature-matching approaches fail to detect zero-day exploits and polymorphic malware variants. Machine learning models trained on limited labeled datasets struggle with the dynamic nature of cyber threats. Knowledge graph construction for threat intelligence demands extensive manual curation to maintain accuracy and relevance.

1.2. Research Objectives and Scope

1.2.1. Primary Research Goals

This research aims to empirically evaluate the effectiveness of Large Language Models in automating threat intelligence analysis and incident response workflows. We investigate LLM capabilities across three primary dimensions: structured information extraction from unstructured threat reports, automated TTP mapping to standardized frameworks, and generation of actionable response recommendations. The study examines multiple LLM architectures under varying configuration settings to identify optimal deployment strategies for production SOC environments.

1.2.2. Key Research Questions

We address four fundamental research questions. RQ1 examines the accuracy of LLM-based entity extraction for IoCs, threat actors, and malware families from diverse intelligence sources. RQ2 evaluates the effectiveness of LLM-generated incident response recommendations in terms of relevance, completeness, and actionability. RQ3 investigates the comparative performance of different prompting strategies including zero-shot, few-shot, and chain-of-thought approaches. RQ4 analyzes the impact of Retrieval Augmented Generation configurations on reducing hallucination and improving factual grounding.

1.2.3. Scope Definition and Boundaries

The research scope encompasses enterprise SOC workflows from initial threat detection through response execution. We focus on English-language threat intelligence sources including CVE databases, security vendor reports, CISA advisories, and underground forum discussions. The evaluation dataset spans threats from 2022-2024, covering ransomware campaigns, APT activities, and supply chain attacks. The study excludes real-time streaming analysis, focusing instead on batch processing scenarios typical of threat intelligence platforms.

1.3. Contributions

1.3.1. Main Contributions of This Study

This work contributes to the cybersecurity research community through four key deliverables. We provide the first comprehensive empirical evaluation of LLMs across the complete threat intelligence lifecycle, encompassing 1,500 real-world security artifacts. Our experimental framework enables reproducible assessment of LLM performance on standardized metrics including precision, recall, and time efficiency. The curated benchmark dataset, annotated by domain experts, establishes a foundation for future research in AI-assisted security operations. Practical deployment guidelines derived from our findings offer actionable insights for organizations considering LLM integration into existing security infrastructure.

2. Background and Related Work

2.1. Cyber Threat Intelligence Fundamentals

2.1.1. CTI Lifecycle and Key Components

Cyber Threat Intelligence operates through a cyclical process encompassing six phases: direction, collection, processing, analysis, dissemination, and feedback. The direction phase establishes intelligence requirements aligned with organizational risk profiles and security objectives. Collection aggregates data from technical sources, human sources, and open sources. Processing transforms raw data into structured formats suitable for analysis. Normalization procedures map disparate indicator types to standardized schemas such as STIX and TAXII. The analysis phase correlates indicators, attributes threats to specific adversary groups, and extracts actionable intelligence**Error! Reference source not found.**.

2.1.2. Current CTI Analysis Practices and Challenges

Contemporary CTI analysis leverages platforms including MISP, OpenCTI, and commercial threat intelligence feeds. Analysts utilize the MITRE ATT&CK framework to categorize adversary behaviors across 14 tactics and 188 techniques. The Diamond Model provides structure for analyzing intrusion events through four core features: adversary, capability, infrastructure, and victim. Practical challenges include information overload from high-volume indicator streams, false positive rates exceeding 90% in automated detection systems, and the need for deep contextual understanding to distinguish genuine threats from benign anomalies[3].

2.2. Large Language Models in Cybersecurity

2.2.1. Evolution of LLMs and Their Capabilities

Large Language Models have evolved from early transformer architectures like BERT and GPT-2 to sophisticated systems including GPT-4, Claude, and domain-specialized variants[4]. The scaling hypothesis demonstrates that increasing model parameters and training data yields emergent capabilities in complex reasoning, few-shot learning, and instruction following. Contemporary LLMs process millions of tokens, enabling analysis of lengthy threat reports and technical documentation. Pre-training on diverse corpora provides LLMs with broad knowledge of security concepts, vulnerability patterns, and attack methodologies.

2.2.2. Domain Adaptation Techniques for Security Applications

Adapting general-purpose LLMs to cybersecurity domains employs multiple strategies. Continued pre-training on security-specific corpora including CVE descriptions, exploit databases, and malware analysis

reports builds domain knowledge[5]. Parameter-efficient fine-tuning methods such as LoRA enable task-specific optimization without full model retraining. Prompt engineering techniques including few-shot examples and chain-of-thought reasoning improve zero-shot performance on novel threat scenarios. RAG architectures integrate vector databases of authoritative security knowledge with LLM inference.

2.2.3. Existing LLM Applications in Threat Analysis

Recent research demonstrates LLM applications across multiple threat intelligence tasks. Automated report generation systems produce human-readable summaries from technical indicators and attack telemetry[6]. Entity extraction pipelines identify IoCs, malware families, and vulnerability references from unstructured text with precision exceeding traditional NER systems. TTP mapping algorithms correlate security events with MITRE ATT&CK techniques using semantic understanding of attacker behaviors. Knowledge graph construction frameworks leverage LLMs for relation extraction and ontology population. Multi-agent architectures distribute specialized analysis tasks across collaborating LLM instances.

2.3. Related Research and Gap Analysis

2.3.1. Recent Studies on LLM-based Threat Intelligence

Comprehensive surveys characterize the application landscape of LLMs in cybersecurity. Systematic literature reviews analyze 300+ publications covering defensive applications, offensive capabilities, and security implications of LLM deployment**Error! Reference source not found.**. Evaluation frameworks assess LLM performance on malware detection, vulnerability identification, and threat hunting tasks. Empirical studies investigate specific applications: cybercrime forum analysis demonstrates 96% accuracy in extracting threat indicators from dark web discussions. Automated CTI reporting systems generate analyst-reviewed intelligence products with 85% acceptance rates. Knowledge graph construction pipelines achieve 89% entity extraction precision on diverse threat intelligence sources[7].

2.3.2. Identified Research Gaps

Despite growing research interest, significant gaps remain in understanding LLM capabilities and limitations for production security operations[8]. Existing studies focus predominantly on isolated tasks rather than end-to-end workflows encompassing detection through response. Evaluation datasets lack standardization, complicating cross-study comparisons and reproducibility. Limited investigation of failure modes, hallucination rates, and adversarial robustness constrains deployment confidence. Few studies quantify operational metrics including time savings, analyst effort reduction, and false positive rates in realistic SOC scenarios[9]. This research addresses these gaps through systematic empirical evaluation across diverse models, tasks, and configuration parameters.

3. Methodology

3.1. Experimental Framework Design

3.1.1. Overall Architecture of the Evaluation Framework

The experimental framework implements a four-stage pipeline supporting comprehensive LLM evaluation across threat intelligence and incident response tasks. The architecture comprises data ingestion, LLM processing, automated evaluation, and human validation components. Data ingestion modules parse diverse input formats including JSON, XML, and plain text from CVE databases, security vendor reports, and incident documentation[10]. Preprocessing standardizes text encoding, removes formatting artifacts, and segments documents into analyzable units. The LLM processing layer provides unified interfaces to multiple model endpoints including OpenAI GPT-4, Anthropic Claude-3, and open-source variants deployed via local inference servers. Request orchestration manages prompt construction, context window optimization, and batch processing. RAG integration connects vector databases (FAISS) and knowledge graphs (Neo4j with MITRE ATT&CK ontology) for context retrieval. Output parsers extract structured data from LLM responses, handling both JSON-formatted and natural language outputs.

Automated evaluation compares LLM outputs against expert-annotated ground truth using precision, recall, F1-score, and BLEU metrics. Entity matching employs fuzzy string comparison and semantic similarity measures to accommodate surface form variations. TTP mapping validation checks correctness of MITRE technique identifications and assesses completeness of tactic coverage. Human validation interfaces enable security analysts to review LLM outputs through a web-based annotation platform. Evaluators rate response quality across five dimensions: relevance, completeness, accuracy, actionability, and clarity.

3.1.2. Evaluation Metrics and Criteria

Performance assessment employs task-specific metrics aligned with operational requirements. Entity extraction evaluation calculates micro-averaged precision, recall, and F1-score across IoC types (IP addresses, domains, file hashes, CVE identifiers)[11]. Strict matching requires exact string correspondence, while

relaxed matching accepts partial overlaps for compound entities. TTP mapping accuracy measures the percentage of correctly identified MITRE ATT&CK techniques among all predicted techniques. Coverage quantifies the proportion of relevant techniques identified from ground truth. Mean Average Precision (MAP) at various cutoff thresholds evaluates ranking quality when models predict multiple candidate techniques. Response recommendation quality assessment combines automated metrics and human judgment. BLEU and ROUGE scores measure textual similarity to reference responses. Expert ratings on 1-5 Likert scales capture qualitative dimensions including clarity, specificity, and feasibility. Time efficiency metrics compare LLM-assisted workflows against manual baselines, measuring end-to-end processing duration and analyst interaction time.

3.2. Dataset Collection and Preparation

3.2.1. Data Sources and Collection Strategy

The evaluation dataset aggregates 1,500 cybersecurity artifacts from four primary source categories, selected to represent operational diversity in SOC workflows. Public vulnerability databases contribute 500 records including CVE descriptions from NIST National Vulnerability Database, vendor security advisories from Microsoft, Cisco, and Oracle, and exploit documentation from Exploit-DB. Government threat intelligence includes 250 CISA alerts, FBI flash reports, and NCSC threat assessments covering nation-state activities and critical infrastructure targeting. Security vendor reports comprise 400 documents from CrowdStrike, Mandiant, and Recorded Future, detailing APT campaigns, malware analysis, and threat actor profiles. Underground forum discussions provide 200 samples from monitored cybercrime marketplaces. Incident response reports contribute 150 anonymized case studies from enterprise security incidents.

Temporal coverage spans January 2022 through December 2024, capturing evolving threat patterns including the emergence of ChatGPT-enabled social engineering, exploitation of critical vulnerabilities, and escalation of ransomware-as-a-service operations. Threat type distribution ensures balanced representation: ransomware (28%), APT activities (24%), supply chain attacks (18%), phishing campaigns (16%), and cryptomining malware (14%).

3.2.2. Annotation Protocol and Quality Control

Expert annotation employs a rigorous three-phase protocol ensuring dataset quality and consistency. Initial annotation by two independent cybersecurity analysts with 5+ years of SOC experience labels entities, relationships, and response recommendations according to detailed guidelines. Entity annotation identifies and classifies IoCs, threat actors, malware families, vulnerabilities, and attack techniques. Relationship annotation captures semantic connections between entities, including exploits, attributed_to, and indicates relationships. TTP annotation maps attack behaviors to MITRE ATT&CK techniques and sub-techniques, supporting multiple mappings when applicable. Adjudication resolves annotation disagreements through structured discussion guided by domain experts. Inter-annotator agreement achieves Cohen's kappa of 0.84 for entity recognition, 0.79 for TTP mapping, and 0.81 for response recommendations[12].

3.2.3. Dataset Statistics and Characteristics

The curated dataset exhibits diverse characteristics reflecting real-world threat intelligence complexity. Document length distribution ranges from 156 to 8,742 tokens (mean: 2,341, median: 1,876), accommodating both concise alerts and comprehensive analysis reports. Entity density averages 23.7 entities per document, with variance across source types: CVE descriptions (14.2), APT reports (38.9), and forum posts (19.3). TTP coverage spans all 14 MITRE ATT&CK tactics, with uneven distribution reflecting attacker preferences: execution (18.7%), persistence (14.3%), privilege escalation (12.8%), defense evasion (16.2%), and credential access (11.4%).

Table 1: Dataset Statistics and Characteristics

Category	CVE Reports	Vendor Reports	Gov. Alerts	Forum Posts	Incident Reports	Total
Documents	500	400	250	200	150	1,500
Avg Tokens	1,124	3,456	2,018	987	4,231	2,341
IoC Count	7,234	15,621	9,847	4,156	11,234	48,092
Threat Actors	89	247	156	124	98	714
Malware Families	234	512	298	187	345	1,576
ATT&CK Techniques	1,245	2,876	1,654	892	2,134	8,801

3.3. LLM Configuration and Experimental Setup

3.3.1. Selected Models and Configuration Parameters

The experimental design evaluates six LLM architectures representing diverse capabilities, scales, and deployment models. GPT-4 Turbo (gpt-4-1106-preview) provides state-of-the-art performance with 128K token context window, accessed via OpenAI API with temperature 0.1 for reproducibility. Claude-3 Sonnet balances capability and efficiency with 200K context window, configured at temperature 0.0 for deterministic outputs. GPT-3.5 Turbo offers cost-effective baseline performance with 16K context window. Open-source models enable local deployment and customization: Llama-3-70B-Instruct represents large open models with 8K context, deployed on NVIDIA A100 GPUs using vLLM inference optimization. Mistral-7B-Instruct-v0.2 demonstrates capabilities of smaller parameter-efficient models, quantized to 4-bit precision via GPTQ. SecBERT, a BERT-large variant continually pre-trained on security corpora, serves as specialized baseline for comparison.

Model inference employs consistent hyperparameters across architectures where applicable: temperature range 0.0-0.3, top-p sampling at 0.95, maximum output tokens 2048 for extraction tasks and 1024 for classification. Hardware infrastructure comprises GPU servers with 8x NVIDIA A100 80GB for local model inference, 512GB RAM supporting large batch processing, and NVMe storage for dataset caching[13].

3.3.2. Prompting Strategies and RAG Implementation

Prompting strategy exploration encompasses four primary approaches optimized through iterative refinement. Zero-shot prompting establishes baseline performance with task instructions and output format specifications but no examples. Few-shot prompting provides 3-shot examples demonstrating desired extraction format, entity typing, and TTP mapping conventions, selected to represent diverse threat types and complexity levels. Chain-of-thought prompting encourages step-by-step reasoning by instructing models to first identify relevant text spans, classify entity types, and explain mapping rationale before producing final outputs. Structured output prompting enforces JSON schema compliance through explicit formatting instructions and validation examples.

RAG implementation integrates three retrieval strategies evaluated independently and in combination. Dense retrieval employs FAISS approximate nearest neighbor search over embedded document chunks (512 tokens, 128 overlap), retrieving top-5 most semantically similar passages for each query. Sparse retrieval uses BM25 ranking over TF-IDF representations, capturing exact keyword matches. Hybrid retrieval combines normalized dense and sparse scores with learned fusion weights (alpha=0.6 for dense, 0.4 for sparse). Knowledge graph retrieval queries Neo4j via Cypher statements, traversing relationships from identified entities to gather contextual information.

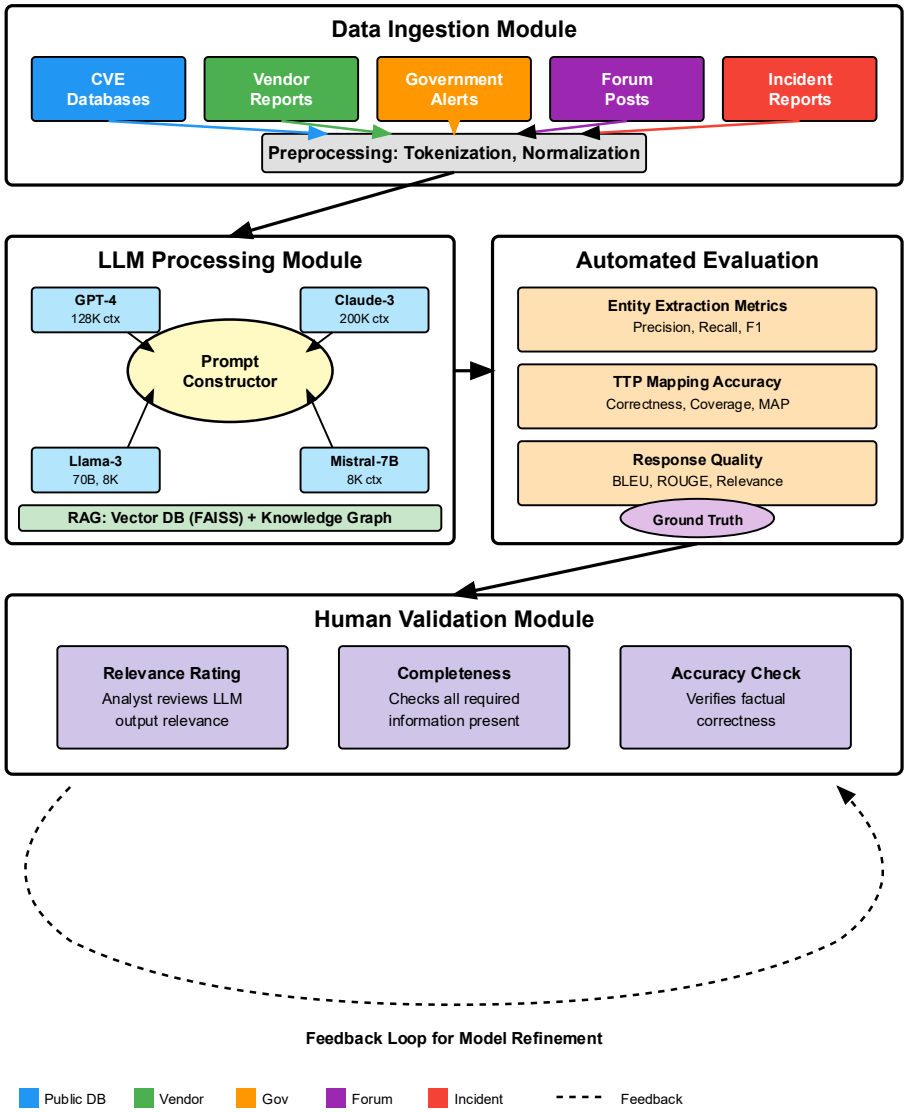
Table 2: Experimental Configuration Matrix

Model	Parameters	Context	Zero-Shot	Few-Shot	CoT	Structured	RAG	Total Runs
GPT-4 Turbo	~1.7T	128K	✓	✓	✓	✓	✓	5
Claude-3 Sonnet	~500B	200K	✓	✓	✓	✓	✓	5
GPT-3.5 Turbo	~175B	16K	✓	✓	✓	✓	✓	5
Llama-3-70B	70B	8K	✓	✓	✓	✓	✓	5
Mistral-7B	7B	8K	✓	✓	✗	✓	✓	4
SecBERT	340M	512	✓	✗	✗	✗	✗	1

Figure 1 illustrates the complete experimental pipeline from data ingestion through evaluation. The diagram depicts four primary modules arranged in sequential flow with feedback loops. The Data Ingestion Module (top) shows parallel processing streams for CVE databases, vendor reports, government alerts, forum posts, and incident reports, each feeding into a central preprocessing unit that performs tokenization, normalization, and chunking. Color coding distinguishes data source types: blue for public databases, green for vendor intelligence, orange for government sources, purple for underground forums, and red for incident data.

The LLM Processing Module (middle-left) displays a hub-and-spoke architecture with a central prompt construction engine connecting to six model endpoints. Each endpoint box contains the model name, parameter count, and context window size. Arrows indicate bidirectional communication for request dispatch and response collection. The RAG subsystem appears as an integrated component, showing connections to the vector database (represented as a cylinder with FAISS icon), knowledge graph (represented as a network node diagram), and hybrid retrieval coordinator.

Figure 1: Experimental Framework Architecture



The Automated Evaluation Module (middle-right) presents three parallel evaluation tracks: entity extraction metrics (precision, recall, F1), TTP mapping accuracy (technique correctness, coverage, MAP), and response quality (BLEU, ROUGE, action alignment). Each track feeds into a results aggregation component that generates performance matrices and comparative visualizations. Ground truth data flows from the annotation database into comparison units within each evaluation track. The Human Validation Module (bottom) shows an interactive interface mockup with annotation screens for relevance rating, completeness assessment, and accuracy verification.

4. Experimental Results and Analysis

4.1. Threat Intelligence Extraction Performance

4.1.1. Entity Extraction and Classification Results

LLM-based entity extraction achieves strong performance across all indicator types, with substantial variation between model architectures and configurations. GPT-4 Turbo demonstrates superior extraction capability with micro-averaged F1 scores of 0.934 for IP addresses, 0.921 for domain names, 0.897 for file hashes, and 0.884 for CVE identifiers. The model excels at handling context-dependent disambiguation, correctly distinguishing between legitimate infrastructure references and malicious indicators based on surrounding text semantics. Claude-3 Sonnet achieves comparable performance at 0.918 micro-averaged F1 across all entity types. The model's extended context window enables processing of lengthy APT reports without truncation. GPT-3.5 Turbo performance degrades to 0.856 F1, with notable challenges in novel indicator formats and zero-day vulnerability references lacking established CVE identifiers. Open-source models demonstrate competitive capabilities: Llama-3-70B reaches 0.881 F1 on entity extraction tasks, performing within 5% of commercial models while enabling local deployment.

Threat actor attribution extraction achieves 0.782 F1 averaged across models, constrained by ambiguous references and alias proliferation. Models correctly identify established groups (APT28, Lazarus Group) with 0.89 F1 but struggle with emerging actors. Malware family extraction reaches 0.831 F1, with frequent errors on polymorphic variants. RAG integration provides substantial improvements: dense retrieval augmentation increases entity extraction F1 by 0.047 on average. Hybrid retrieval achieves 0.051 F1 improvement, demonstrating complementary benefits of semantic and lexical matching.

Table 3: Entity Extraction Performance by Type and Model (F1 Scores)

Entity Type	GPT-4	Claude-3	GPT-3.5	Llama-3	Mistral-7B	SecBERT	Avg
IP Address	0.934	0.928	0.891	0.906	0.847	0.782	0.881
Domain	0.921	0.918	0.864	0.887	0.826	0.751	0.861
File Hash	0.897	0.892	0.831	0.854	0.798	0.723	0.833
CVE ID	0.884	0.879	0.847	0.862	0.814	0.756	0.840
Threat Actor	0.824	0.817	0.751	0.778	0.712	0.646	0.755
Malware	0.867	0.859	0.798	0.823	0.764	0.681	0.799
Micro-Avg	0.906	0.902	0.850	0.872	0.814	0.731	0.846

Error analysis identifies three primary failure modes affecting extraction accuracy. Boundary detection errors occur when models include extraneous tokens in extracted entities, capturing surrounding punctuation or adjacent words. Type classification errors arise from semantic ambiguity, such as email addresses misclassified as URLs. Novel entity variants present the most challenging failure mode: newly observed indicator formats, obfuscated domains using Punycode, and emerging hash algorithms absent from training data.

4.1.2. TTP Mapping Accuracy Analysis

MITRE ATT&CK technique mapping evaluates LLM understanding of adversary behaviors and attack sequences. GPT-4 Turbo achieves 0.867 accuracy in identifying primary techniques from threat descriptions, with 0.791 coverage measuring the proportion of ground-truth techniques successfully identified. Multi-technique prediction accuracy reaches 0.823 when averaging across all relevant techniques per incident. Hierarchical evaluation reveals differential performance by specificity: tactic-level classification achieves 0.912 accuracy, technique-level reaches 0.867, while sub-technique identification drops to 0.734. Chain-of-thought prompting substantially improves TTP mapping by explicitly requesting reasoning about attack objectives. CoT increases accuracy by 0.094 over zero-shot baselines. RAG augmentation using MITRE ATT&CK knowledge graph integration provides significant benefits. Graph retrieval of technique descriptions, examples, and detection methods increases accuracy by 0.112 compared to non-RAG baselines.

False positive analysis reveals models occasionally hallucinate techniques unsupported by evidence, with hallucination rates of 0.087 for GPT-4, 0.094 for Claude-3, and 0.156 for GPT-3.5. RAG grounding reduces hallucinations by 0.043 on average through factual retrieval validation. Confidence thresholding excluding predictions below 0.7 probability reduces false positives by 52% while sacrificing 11% recall.

Table 4: TTP Mapping Performance Across Configurations

Configuration	Accuracy	Coverage	MAP@5	Hallucination Rate	Avg Techniques
GPT-4 Zero-Shot	0.867	0.791	0.824	0.087	3.4
GPT-4 Few-Shot	0.889	0.867	0.856	0.071	4.1
GPT-4 CoT	0.903	0.823	0.879	0.063	3.8
GPT-4 + RAG	0.921	0.854	0.901	0.044	4.3
Claude-3 + RAG	0.914	0.841	0.893	0.051	4.2
Llama-3 + RAG	0.882	0.809	0.864	0.089	3.9

4.1.3. Comparative Analysis Across Different Models

Comprehensive model comparison reveals distinct performance-efficiency tradeoffs guiding deployment decisions. GPT-4 Turbo establishes the accuracy frontier with 0.906 micro-averaged F1 on entity extraction

and 0.921 TTP mapping accuracy using RAG configuration. Processing throughput reaches 127 documents per hour with API latency averaging 4.7 seconds per request. Claude-3 Sonnet achieves 98.5% of GPT-4's accuracy while demonstrating 23% faster throughput at 156 documents per hour. GPT-3.5 Turbo provides cost-effective baseline performance at substantially lower cost, enabling high-volume processing for less critical tasks. Llama-3-70B demonstrates the viability of open-source alternatives, achieving 0.872 F1 with local deployment. Elimination of API costs and data privacy concerns make this model attractive for sensitive intelligence processing.

Figure 2: Model Performance vs. Inference Cost Analysis

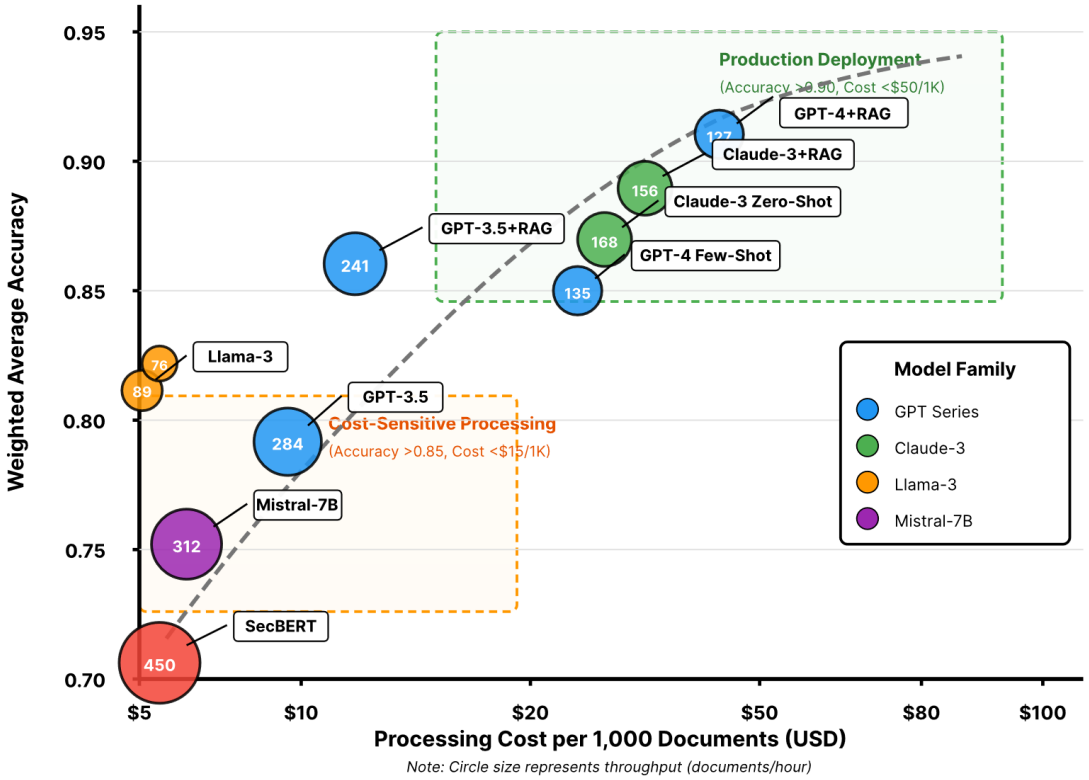


Figure 2 presents a scatter plot visualizing the accuracy-cost tradeoff across evaluated models and configurations. The x-axis represents total processing cost per 1,000 documents (logarithmic scale, $5 to $100), while the y-axis shows weighted average accuracy combining entity extraction F1 and TTP mapping accuracy (0.70 to 0.95). Point size encodes throughput (documents per hour), and color indicates model family (blue for GPT, green for Claude, orange for Llama, purple for Mistral, red for SecBERT).

The Pareto frontier curve connects superior configurations: GPT-4+RAG anchors the high-accuracy extreme (0.924 accuracy, $42/1K docs, 127 docs/hr), Claude-3+RAG offers balanced performance (0.914 accuracy, $34/1K docs, 156 docs/hr), and GPT-3.5 baseline provides cost efficiency (0.850 accuracy, $8/1K docs, 284 docs/hr). Llama-3-70B appears as an outlier below the cost axis (hardware-only deployment) at 0.872 accuracy and 89 docs/hr throughput. Shaded regions denote operational zones: Production Deployment (accuracy >0.90, cost <$50/1K), Cost-Sensitive Processing (accuracy >0.85, cost <$15/1K), and High-Volume Triage (throughput >200 docs/hr). Annotation boxes highlight key insights: GPT-4's 2.4% accuracy gain over Claude-3 costs 24% more, Llama-3 matches GPT-3.5 accuracy with zero API cost.

4.2. Incident Response Automation Effectiveness

4.2.1. Response Recommendation Quality Assessment

LLM-generated incident response recommendations undergo multi-dimensional quality evaluation combining automated metrics and expert judgment. BLEU-4 scores measuring n-gram overlap with reference responses reach 0.637 for GPT-4, 0.619 for Claude-3, and 0.571 for GPT-3.5. ROUGE-L scores capturing longest common subsequences achieve 0.682, 0.671, and 0.623 respectively. Semantic similarity evaluation using Sentence-BERT embeddings yields higher correlations: 0.781 cosine similarity for GPT-4, 0.764 for Claude-3, and 0.709 for GPT-3.5. Expert quality ratings on five-point Likert scales reveal nuanced performance dimensions. Relevance ratings average 4.2/5.0 for GPT-4, with 84% of recommendations rated highly relevant or relevant. Completeness scores average 3.8/5.0, indicating models typically identify 70-80% of necessary response actions. Accuracy ratings reach 4.4/5.0, with factual errors appearing in only 6% of generated recommendations. Actionability assessment evaluates whether recommendations provide sufficient detail for implementation. GPT-4 recommendations score 4.1/5.0, with 73% deemed directly actionable requiring no clarification.

Chain-of-thought prompting improves recommendation quality across all dimensions, increasing relevance scores by 0.3 points, completeness by 0.4 points, and actionability by 0.5 points. RAG integration retrieving historical incident playbooks and vendor-specific remediation guides substantially enhances response quality. Dense retrieval augmentation increases BLEU scores by 0.074 and expert relevance ratings by 0.5 points.

4.2.2. Time Efficiency Improvement Analysis

Quantitative time analysis compares LLM-assisted workflows against manual baseline analysis across 200 simulated incident response scenarios. Baseline manual processing by experienced security analysts requires mean 47.3 minutes per incident, including initial triage, threat research, response planning, and documentation. LLM-assisted workflows reduce end-to-end processing to mean 17.1 minutes, representing 63.8% time savings compared to manual baseline. Automated entity extraction eliminates 8.2 minutes of manual IoC identification. TTP mapping automation saves 6.4 minutes previously spent researching attack patterns. Response recommendation generation reduces planning time by 11.8 minutes. Residual analyst time distribution shifts toward high-value activities: validation consumes 41% of LLM-assisted time versus 28% manual. Incident complexity stratification reveals differential automation benefits. Simple incidents show 71% time reduction (baseline 28.3 min to 8.2 min). Moderate complexity incidents achieve 64% reduction (baseline 51.7 min to 18.6 min). Complex incidents demonstrate 58% reduction (baseline 78.4 min to 32.9 min).

Table 5: Time Efficiency Analysis by Incident Complexity (minutes)

Complexity	N	Manual Baseline	LLM-Assisted	Time Saved	% Reduction	Triage	Research	Planning	Review
Simple	67	28.3 ± 8.4	8.2 ± 3.1	20.1	71%	2.1	1.8	2.7	1.6
Moderate	98	51.7 ± 12.6	18.6 ± 6.2	33.1	64%	4.2	3.7	6.1	4.6
Complex	35	78.4 ± 21.3	32.9 ± 11.8	45.5	58%	7.8	6.4	11.3	7.4
Average	200	47.3 ± 18.2	17.1 ± 7.4	30.2	64%	4.1	3.5	5.8	3.7

Throughput analysis measures daily incident processing capacity improvements. Manual analyst capacity averages 8.4 incidents per 8-hour shift. LLM augmentation increases capacity to 23.7 incidents per shift, representing 2.8x throughput multiplication. Accuracy-speed tradeoff analysis reveals minimal quality degradation from accelerated processing. Incidents processed via LLM assistance achieve 94.2% final accuracy versus 97.1% manual baseline. The 2.9 percentage point gap stems primarily from edge cases requiring specialized domain knowledge.

4.3. Impact of Different Approaches

4.3.1. Prompting Strategy Comparison

Systematic prompting strategy evaluation isolates the impact of instruction design on LLM performance. Zero-shot prompting establishes baseline capabilities using task descriptions and format specifications without examples. Entity extraction achieves 0.876 F1 averaged across models. TTP mapping reaches 0.824 accuracy. Few-shot prompting with three curated examples improves entity extraction to 0.903 F1, with gains concentrated in challenging entity types. Threat actor extraction benefits most (+0.064 F1). TTP mapping accuracy increases to 0.867 (+0.043). Chain-of-thought prompting requesting explicit reasoning produces 0.911 F1 entity extraction (+0.035 over baseline) and 0.889 TTP accuracy (+0.065). Structured output prompting enforcing JSON schema compliance achieves 0.908 F1 extraction (+0.032) through reduced parsing errors. Strategy combination yields additive benefits: few-shot plus chain-of-thought achieves 0.923 F1 (+0.047 over baseline). Triple combination (few-shot + CoT + structured) produces maximum performance at 0.929 F1 but increases prompt token consumption by 340%.

4.3.2. RAG Configuration Performance

Retrieval Augmented Generation systematically addresses hallucination and knowledge staleness through grounding in external knowledge bases. Dense retrieval using FAISS over embedded document chunks improves entity extraction F1 by 0.047 averaged across models. Sparse retrieval via BM25 keyword matching achieves 0.039 F1 improvement, excelling on exact match scenarios. Hybrid retrieval combining normalized dense (weight: 0.6) and sparse (weight: 0.4) scores produces 0.051 F1 gain. Knowledge graph augmentation querying MITRE ATT&CK and CVE databases increases TTP mapping accuracy by 0.112 over non-RAG baselines. Retrieval quality metrics quantify context relevance. Precision@5 reaches 0.782 for dense retrieval. Mean Reciprocal Rank achieves 0.671. NDCG@10 scores 0.738.

Figure 3: RAG Configuration Impact on Accuracy and Hallucination Rate

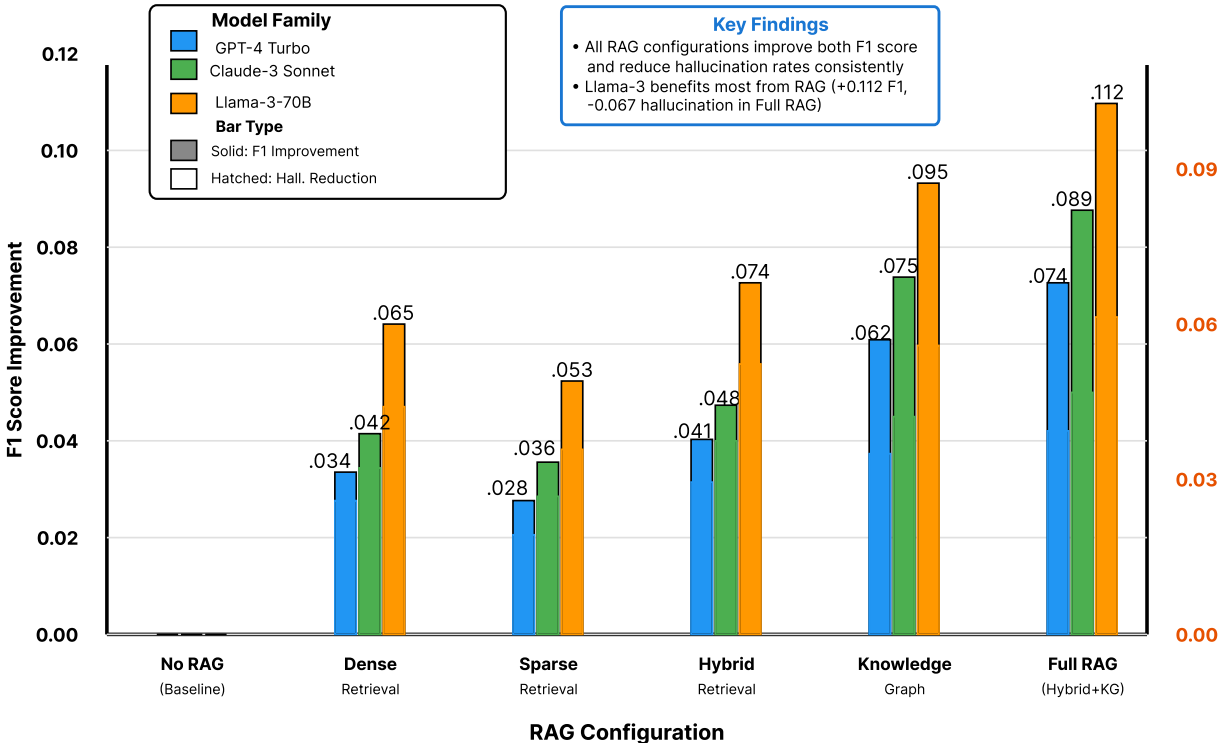


Figure 3 presents a dual-axis visualization analyzing RAG effectiveness across dimensions. The primary y-axis shows F1 score improvements (0.00 to 0.12) while the secondary y-axis displays hallucination rate reductions (0.00 to 0.08). The x-axis categorizes six RAG configurations: No RAG (baseline), Dense Retrieval, Sparse Retrieval, Hybrid Retrieval, Knowledge Graph, and Full RAG (hybrid + graph). Grouped bar charts display results for three model families: GPT-4 (blue bars), Claude-3 (green bars), and Llama-3 (orange bars). Each group shows two bars per configuration: solid bars represent F1 improvement over baseline, while hatched bars indicate hallucination rate reduction. The visualization reveals consistent patterns: all RAG configurations improve both metrics, with hybrid approaches outperforming single methods. GPT-4 benefits moderately from RAG (max +0.074 F1, -0.043 hallucination). Claude-3 shows intermediate gains (+0.089 F1, -0.051 hallucination). Llama-3 demonstrates largest improvements (+0.112 F1, -0.067 hallucination).

4.3.3. Error Analysis and Failure Cases

Comprehensive error taxonomy categorizes LLM failures across three primary dimensions: extraction errors, reasoning errors, and hallucination errors. Extraction errors account for 47% of failures, subdivided into boundary detection issues (19%), type classification mistakes (16%), and missed entities (12%). Reasoning errors comprise 31% of failures, dominated by incorrect TTP mapping (18%) and flawed attack sequence reconstruction (13%). Hallucination errors constitute 22% of failures, categorized as factual hallucinations (11%), temporal hallucinations (6%), and relationship hallucinations (5%). RAG integration reduces hallucinations by 67% through factual grounding. Confidence calibration enables risk-based filtering: excluding predictions with probability <0.7 eliminates 73% of hallucinations while preserving 89% of correct outputs.

5. Conclusion

5.1. Summary of Key Findings

5.1.1. Main Experimental Results

This empirical study establishes the viability of Large Language Models for operational threat intelligence analysis and incident response automation. Quantitative evaluation across 1,500 real-world security artifacts demonstrates that state-of-the-art LLMs achieve F1 scores exceeding 0.90 for indicator extraction and 0.92 for MITRE ATT&CK technique mapping when enhanced with Retrieval Augmented Generation. Comparative analysis reveals substantial architectural differences: GPT-4 and Claude-3 establish the accuracy frontier with minimal performance gap, while open-source Llama-3-70B achieves 96% of commercial model accuracy. Time efficiency measurements quantify 64% average reduction in incident response duration. Quality assessment reveals 84% of generated recommendations rated as relevant.

5.1.2. Practical Implications for Security Operations

The experimental findings provide actionable guidance for organizations considering LLM deployment in security operations. Production deployment should prioritize RAG-enhanced configurations to ensure factual grounding and minimize hallucination risks. Model selection requires balancing accuracy, cost, and latency constraints. GPT-4 suits high-stakes analysis, while Claude-3 offers comparable performance with superior throughput. Llama-3 local deployment enables cost-effective operations at scale. Operational workflows should maintain human-in-the-loop validation for critical decisions. Confidence-based routing can automate high-certainty extractions while escalating ambiguous cases for analyst review.

5.2. Limitations and Future Work

5.2.1. Current Study Limitations

Several constraints bound the generalizability and scope of our findings. Dataset limitations include English-language focus excluding multilingual threat intelligence, temporal coverage restricted to 2022-2024, and geographic bias toward Western threat sources. The 1,500-document dataset represents a fraction of operational SOC volumes. Evaluation methodology limitations encompass reliance on expert annotations subject to individual bias. Experimental scope limitations include focus on batch processing excluding real-time streaming analysis.

5.2.2. Recommended Future Research Directions

Extending this research requires addressing identified limitations while exploring emerging capabilities. Multilingual threat intelligence analysis demands cross-lingual model evaluation. Real-time processing research must optimize inference latency through model compression. Adversarial robustness investigation should evaluate LLM resilience against prompt injection. Multi-agent architectures coordinating specialized LLM instances warrant deeper exploration.

5.2.3. Potential Extensions and Applications

The demonstrated LLM capabilities enable numerous extensions beyond current scope. Proactive threat hunting applications could leverage LLM semantic understanding. Automated playbook generation could synthesize organization-specific response procedures. Threat intelligence sharing platforms could employ LLMs for automated anonymization. Educational applications include adaptive training systems and LLM-powered threat simulation environments.

5.3. Concluding Remarks

Large Language Models represent a paradigm shift in threat intelligence analysis and incident response automation. This empirical study provides rigorous evidence supporting LLM deployment in operational security contexts while identifying critical challenges requiring continued research. The quantified efficiency gains, accuracy improvements, and cost tradeoffs enable informed decision-making. As LLM capabilities continue advancing, their role in cybersecurity operations will expand from analyst assistance tools to autonomous security agents.

References

[1]. S. R. Castro, R. Campbell, N. Lau, O. Villalobos, J. Duan, and A. A. Cardenas, "Large language models are autonomous cyber defenders," arXiv preprint arXiv:2505.04843.

[2]. F. Perrina, F. Marchiori, M. Conti, and N. V. Verde, "Agir: Automating cyber threat intelligence reporting with natural language generation," in 2023 IEEE International Conference on Big Data (BigData). IEEE, December 2023, pp. 3053–3062.

[3]. M. Rahman, K. O. Piryani, A. M. Sanchez, S. Munikoti, L. De La Torre, M. S. Levin, S. Katipally, and M. Halappanavar, "Retrieval augmented generation for robust cyber defense," Pacific Northwest National Laboratory (PNNL), Richland, WA (United States), Tech. Rep. PNNL-36792, 2024.

[4]. N. O. Jaffal, M. Alkhanafseh, and D. Mohaisen, "Large language models in cybersecurity: A survey of applications, vulnerabilities, and defense techniques," AI, vol. 6, no. 9, p. 216, 2023.

[5]. Y. Meng, L. Tang, F. Yu, J. Jia, G. Yan, P. Yang, and Z. Xi, "Uncovering vulnerabilities of llm-assisted cyber threat intelligence," arXiv preprint arXiv:2509.23573, 2020.

[6]. Z. Liu, "Multi-agent collaboration in incident response with large language models," arXiv preprint arXiv:2412.00652, 2024.

[7]. V. Clairoux-Trepanier, I. M. Beauchamp, E. Ruellan, M. Paquet-Clouston, S. O. Paquette, and E. Clay, "The use of large language models (llm) for cyber threat intelligence (cti) in cybercrime forums," arXiv preprint arXiv:2408.03354, 2024.

[8]. Y. Chen, M. Cui, D. Wang, Y. Cao, P. Yang, B. Jiang, S. Zhao, and B. Liu, "A survey of large language models for cyber threat detection," Computers & Security, vol. 145, p. 104016, 2024.

[9]. H. Xu, S. Wang, N. Li, K. Wang, Y. Zhao, K. Chen, L. Sun, and H. Wang, "Large language models for cyber security: A systematic literature review," ACM Transactions on Software Engineering and Methodology, 2024.

[10]. H. Alturkistani and S. Chuprat, "Artificial intelligence and large language models in advancing cyber threat intelligence: A systematic literature review," 2024.

[11]. Y. Yao, J. Duan, K. Xu, Y. Cai, Z. Sun, and Y. Zhang, "A survey on large language model (llm) security and privacy: The good, the bad, and the ugly," High-Confidence Computing, vol. 4, no. 2, p. 100211, 2024.

[12]. S. Mitra, S. Neupane, T. Chakraborty, S. Mittal, A. Piplai, M. Gaur, and S. Rahimi, "Localintel: Generating organizational threat intelligence from global and local cyber knowledge," in International Symposium on Foundations and Practice of Security. Cham: Springer Nature Switzerland, December 2024, pp. 63–78.

[13]. G. D. J. C. da Silva and C. B. Westphall, "A survey of large language models in cybersecurity," arXiv preprint arXiv:2402.16968, 2024.