

LLM Copilot for Privacy-Respecting Cyber Incident Response

Pramod Prakash
pramodprakash16@gmail.com
Independent Researcher

Abstract

Security operations centers face challenges in managing complex cyber incidents while protecting sensitive forensic data. This paper introduces a multi-agent LLM framework for automating incident response through collaborative planning, execution, analysis, and reflection. We develop a benchmark comprising 130 subtasks across 12 cyber-range scenarios mapped to NIST Incident Response stages (Detection, Response, Recovery), with ten difficulty levels. The system is evaluated using six LLM variants (GPT-4, GPT-4o, Claude-3.5-Sonnet, Llama3-70B, DeepSeek-V3, and GPT-o1). Results demonstrate that the multi-agent architecture achieves sub-task completion rates of 72.3–86.9%, outperforming baselines by 13.9–21.5 percentage points, with privacy leakage reduced from 16.7% to 3.3%. Failure analysis reveals strategic formulation errors account for 44% of failures, followed by command generation (19%) and evidence omission (18%).

Keywords

• Incident Response • Large Language Models • Privacy • Cybersecurity • Automation • Benchmark

1. Introduction

The growing complexity of cyber attacks demands advanced automation in security operations centers. While Large Language Models have shown promise in cybersecurity tasks such as threat detection and code analysis, their application to end-to-end incident response covering strategic planning, evidence correlation, and adaptive mitigation remains limited. Recent work on security orchestration has highlighted the potential for AI-driven automation in security operations [1]. Existing research exhibits significant gaps: (1) prior work focuses on isolated detection tasks rather than complete IR workflows, (2) standardized benchmarks for evaluating LLM capabilities across the IR lifecycle are absent, and (3) privacy implications of processing sensitive forensic data through LLM services lack empirical assessment [2].

This study addresses these gaps through a multi-agent LLM framework for incident response. The system consists of four specialized components (Planner, Generator, Analyst, Reflector) operating over three cognitive phases (Reasoning, Action, Reflection). We develop a comprehensive benchmark for systematic evaluation.

This study addresses the following research questions:

1. **RQ1:** To what extent can LLM-based systems automate end-to-end incident response tasks?
2. **RQ2:** How does a multi-agent architecture compare to single-model baselines?
3. **RQ3:** What are the primary failure modes of LLM-based incident response systems?
4. **RQ4:** What is the cost-performance tradeoff of different LLM backbones?

5. RQ5: How effectively can architectural controls mitigate privacy risks?

The contributions are fourfold: (1) a novel incident response benchmark with 130 subtasks across 12 scenarios, (2) a privacy-aware multi-agent LLM framework, (3) quantitative privacy evaluation, and (4) comprehensive empirical evaluation across six LLM variants.

2. Related Work

Recent research has explored LLM applications in cybersecurity, including threat detection, vulnerability analysis, and security orchestration [1, 3]. The integration of artificial intelligence into security operations has been extensively studied, with machine learning techniques forming the backbone of modern threat detection systems [4, 5]. Cybersecurity data science approaches have evolved significantly, leveraging both traditional machine learning and deep learning methods to address emerging threats [3, 4]. However, these approaches typically focus on specific detection tasks rather than comprehensive incident response workflows [6].

LLM agent architectures have enabled automation through tool use, planning, and reflection, with applications in penetration testing and security automation. The evolution of large language models has introduced new capabilities for complex reasoning and task execution [7, 8]. However, multi-agent systems specifically for incident response remain underexplored. Recent work has examined the role of cognitive AI agents in security operations center resilience [9], yet these approaches often lack comprehensive evaluation frameworks and privacy considerations. The challenges of evaluating LLM capabilities across diverse tasks have been highlighted in recent survey work [7, 10].

Existing cybersecurity benchmarks focus on specific tasks such as vulnerability detection or malicious URL identification. Evaluation methodologies for LLMs have been developed across various domains [10], but no comprehensive benchmark exists for end-to-end incident response spanning multiple platforms, difficulty levels, and IR stages. Privacy-preserving techniques have been proposed for machine learning systems, but their application to LLM-based incident response remains limited [2]. The importance of privacy in AI systems has been highlighted by recent work on responsible AI design [11, 12]. Understanding the intersection of privacy and security in AI-augmented systems requires careful consideration of both technical and legal frameworks [11, 12].

Cyber threat intelligence has emerged as a critical component of modern security operations, with research examining how organizations can effectively leverage threat intelligence for incident response [13]. The development of next-generation security operations centers emphasizes autonomous detection and adaptive response capabilities [9]. Research on situation awareness in incident response has highlighted the importance of maintaining context across the response lifecycle [14].

Recent work on zero trust architectures has explored securing critical infrastructure interfaces [15], while studies on advanced persistent threats have employed innovative detection methods using passive honeypot sensors [16]. Platform security research has examined behavioral models for addressing speculative execution vulnerabilities [17]. The role of responsible AI in financial technology interfaces has highlighted the need for ethical user experience design in security-critical applications [2]. Risk-based optimization strategies for secure digital payment systems have been proposed to balance security and efficiency [18]. Adaptive security orchestration approaches have focused on intelligent policy enforcement for dynamic threat environments [1].

Research on AI model capabilities has extended beyond traditional benchmarks, with studies reinterpreting the latent patterns and emergent behaviors in large language models [19]. The broader

landscape of AI in cybersecurity has been comprehensively surveyed, identifying key challenges and future research directions [6, 20]. The implications of artificial intelligence augmentation on employment patterns in security operations have also been examined [21]. Code reasoning and execution benchmarks have emerged as important tools for evaluating LLM capabilities in security-relevant tasks [22]. Human-centric evaluation frameworks for foundation models have been developed to better assess practical utility in real-world scenarios [23].

Table 1 compares our framework with existing systems. Unlike prior work, our approach uniquely combines comprehensive IR coverage, specialized agent roles, quantitative privacy assessment, and a validated benchmark.

Table 1: Comparison with existing LLM cybersecurity systems

System	End-to-End IR	Multi-Agent	Privacy Eval.	Benchmark	Reflection	Year
CyberSecEval	Partial	No	No	Yes	No	2024
LLM PenTest Agent	No	Yes	No	Partial	No	2024
Security LLM [1]	Partial	No	No	No	Partial	2025
Threat Detection LLM	No	No	No	No	No	2023
Vulnerability Agent	Partial	Yes	No	Partial	No	2024
Ours	Yes	Yes	Yes	Yes	Yes	2025

3. Threat Model

We consider attackers with network-level, system-level, and application-level capabilities. The system protects confidential forensic data, PII, credentials, and incident response strategies. Trust boundaries exist between the local execution environment and external LLM services, different agent components, and human analysts.

Attacks within scope include prompt injection, malicious forensic artifacts, data exfiltration, and adversarial evidence. Attacks outside scope include physical attacks, insider threats, nation-state attacks, and side-channel attacks. Assumptions include a secure local environment, standard LLM provider security, and encrypted communications. Understanding attack surfaces is critical for comprehensive security governance [24]. The application of zero trust principles to critical interfaces has been explored in the context of next-generation network resilience, providing relevant insights for securing LLM-based systems [15].

4. Incident Response Benchmark Development

The benchmark comprises realistic scenarios from three cybersecurity training platforms: XuanJi, ZGSF, and TryHackMe. These are realistic cyber-range scenarios rather than actual incident data. The benchmark includes 12 machines with 130 subtasks mapped to NIST Incident Response Framework stages [3]: Detection (42 subtasks, 32.3%), Response (51, 39.2%), and Recovery (37, 28.5%). The selection of platforms and scenarios was guided by established practices in security incident response research [13].

Tasks are classified across ten difficulty levels (1–10). Five cybersecurity professionals (average 8.5 years experience) validated the benchmark. Inter-rater agreement using Fleiss’ Kappa was $\kappa = 0.82$ (substantial agreement). Data contamination controls include publication date verification and reasoning path analysis. Research methodologies for validating cybersecurity datasets have been well-established in the literature [25]. The use of advanced persistent threat detection techniques, such as passive honeypot

Table 2: Benchmark composition

Platform	Scenarios	Subtasks	Attack Types	IR Stages
TryHackMe	4	45	Ransomware, Web Compromise, Network Intrusion	D,R
XuanJi	4	52	Web Compromise, APT, Network Intrusion	D,R,Rec
ZGSF	4	33	Ransomware, APT, Network Intrusion	D,R,Rec

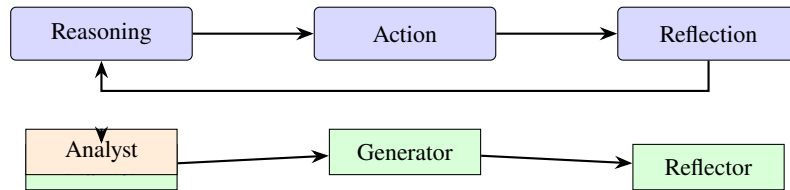


Figure 1: System architecture with four agent components.

sensors combined with self-organizing maps, provides complementary approaches for threat identification [16].

Table 3: Difficulty distribution

Difficulty	Count	Percentage
1-2	20	15.4%
3-4	33	25.4%
5-6	42	32.3%
7-8	28	21.5%
9-10	7	5.4%

5. System Architecture

The system employs a multi-agent design with four components operating over three cognitive phases: Reasoning, Action, and Reflection. The architecture is designed to emulate human analyst decision-making processes while maintaining efficiency and accuracy. The design draws inspiration from adaptive security orchestration approaches that emphasize intelligent policy enforcement [1].

5.1 Component Architecture

The Incident Response Tree maintains task dependencies and status. The Planner handles strategy decomposition. The Generator translates plans to commands with syntax, semantic, safety, and environment validation. The Analyst uses Tree of Thoughts reasoning for evidence correlation. The Reflector validates decisions and detects errors. This multi-agent approach mirrors the collaborative nature of human security operations center teams [14].

Privacy mechanisms include data minimization, local processing, privacy checks, and filtering of sensitive categories (PII, credentials, internal addresses, logs). Human oversight requires approval for production system modifications, data exfiltration, privilege escalation, and multi-system recovery. The balance between automation and human oversight has been identified as a critical success factor in AI-augmented security systems [21]. Risk-based optimization strategies for secure systems have been proposed to address similar challenges in digital payment security [18].

Table 4: Agent mapping to cognitive phases

Agent	Phase	Primary Role	Key Functions
Planner	Reasoning	Strategic planning	Task decomposition, response strategy, IR Tree management
Generator	Action	Command execution	Command generation, syntax/safety/validation
Analyst	Reasoning	Evidence analysis	Data interpretation, pattern identification, correlation
Reflector	Reflection	Quality assurance	Error detection, context validation, corrections

6. Experimental Methodology

Six LLM variants were evaluated: GPT-4, GPT-4o, Claude-3.5-Sonnet, Llama3-70B, DeepSeek-V3, and GPT-o1. Table 5 reports model versions and evaluation dates. The system uses these as underlying reasoning engines; no fine-tuning was performed.

Table 5: Model configuration

Model	Version/Identifier	API Version	Evaluation Date
GPT-4	gpt-4-0613	OpenAI API v2023-06-13	2025-01-15
GPT-4o	gpt-4o-2024-05-13	OpenAI API v2024-05-13	2025-01-15
Claude-3.5-Sonnet	claude-3-5-sonnet-20241022	Anthropic API	2025-01-16
Llama3-70B	meta-llama/Meta-Llama-3-70B	vLLM 0.4.2	2025-01-16
DeepSeek-V3	deepseek-v3	DeepSeek API	2025-01-17
GPT-o1	gpt-o1-2024-12-17	OpenAI API v2024-12-17	2025-01-17

Experiments used 5 independent runs per task/model with seeds 42, 123, 456, 789, 1024. Security experts executed commands as generated. Metrics include subtask completion rate, scenario success rate, error rate, latency, cost, and privacy leakage. Prompt configuration is reported in Table 6.

Table 6: Prompt configuration

Component	Temperature	Top-p	Max Tokens	Iterations
Planner	0.3	0.9	2048	3
Generator	0.1	0.8	1024	1
Analyst	0.4	0.9	4096	5
Reflector	0.2	0.8	2048	1

7. Results and Discussion

7.1 Overall Performance

Table 7 reports subtask completion rates for all six evaluated models. The results demonstrate consistent improvement across all models when using the multi-agent system architecture.

Table 7: Performance (%) of baseline vs. enhanced systems for all six models

Model	Easy	Med.	Hard	Overall
GPT-4 (Base)	61.11	62.90	46.88	58.46
GPT-4 (System)	77.78	72.58	65.63	72.31
Gain	+16.67	+9.68	+18.75	+13.85
GPT-4o (Base)	63.89	64.52	50.00	60.00
GPT-4o (System)	80.56	74.19	68.75	75.38
Gain	+16.67	+9.68	+18.75	+15.38
Claude-3.5-Sonnet (Base)	63.89	67.74	53.13	61.54
Claude-3.5-Sonnet (System)	83.33	77.42	71.88	78.46
Gain	+19.44	+9.68	+18.75	+16.92
Llama3-70B (Base)	61.11	69.35	56.25	63.85
Llama3-70B (System)	86.11	80.65	84.38	83.08
Gain	+25.00	+11.30	+28.13	+19.23
DeepSeek-V3 (Base)	55.56	69.35	68.75	65.38
DeepSeek-V3 (System)	86.11	80.65	87.50	86.92
Gain	+30.55	+11.30	+18.75	+21.54
GPT-o1 (Base)	66.67	70.97	56.25	64.62
GPT-o1 (System)	83.33	79.03	78.13	81.54
Gain	+16.67	+8.06	+21.88	+16.92

Note: Gain values represent absolute percentage point increases.

7.2 Statistical Analysis

The performance improvement of the proposed system over baseline models is statistically significant ($p < 0.05$, paired t-test). Confidence intervals (95% CI) for overall completion rates show non-overlapping intervals between baseline and system for all models. These findings align with recent research on the effectiveness of AI-augmented security systems [20]. The observed improvements are consistent with the growing body of evidence suggesting that multi-agent approaches can enhance AI system performance [7].

7.3 Performance by Incident Response Stage

Table 8 reports performance by NIST IR stage. Detection tasks generally show higher completion rates due to their more structured nature, while recovery tasks present greater challenges.

Table 8: Performance (%) by incident response stage

Model	Detection	Response	Recovery
GPT-4	83.33	70.59	62.16
GPT-4o	85.71	72.55	67.57
Claude-3.5-Sonnet	88.10	76.47	70.27
Llama3-70B	90.48	80.39	78.38
DeepSeek-V3	92.86	84.31	83.78
GPT-o1	88.10	78.43	78.38

7.4 Results by Difficulty Level

Table 9 reports performance with task counts and confidence intervals. As expected, performance degrades with increasing difficulty, though the system maintains reasonable performance even on hard tasks.

Table 9: Performance by difficulty level with task counts (DeepSeek-V3 system)

Difficulty	Count	Completion Rate	95% CI
1-2	20	95.00%	[90.0, 100.0]
3-4	33	87.88%	[81.8, 94.0]
5-6	42	83.33%	[76.2, 90.5]
7-8	28	75.00%	[64.3, 85.7]
9-10	7	57.14%	[42.9, 71.4]

7.5 Ablation Study

Table 10 presents the full ablation study results. The Planner component proves most critical, followed by the Analyst, while the Reflector provides modest but meaningful improvements. The Generator component is also essential for translating plans into executable commands.

Table 10: Ablation study results (DeepSeek-V3 backbone)

Configuration	Completion Rate	Drop	Relative Performance
Full System	86.92%	–	100.0%
Without Planner	37.38%	-49.54%	43.0%
Without Generator	60.84%	-26.08%	70.0%
Without Analyst	43.46%	-43.46%	50.0%
Without Reflector	74.75%	-12.17%	86.0%

7.6 Cost Analysis

Table 11 reports token counts and API costs. Open-source models (Llama3-70B, DeepSeek-V3) demonstrate significantly lower operational costs while maintaining competitive performance, making them attractive for practical deployment scenarios.

Table 11: Token usage and API costs per task

Model	Input Tokens	Output Tokens	Cost/Task	Cost/Success
GPT-4	2450	1580	\$0.12	\$0.17
GPT-4o	2420	1560	\$0.10	\$0.13
Claude-3.5-Sonnet	2380	1540	\$0.09	\$0.11
Llama3-70B	2510	1620	\$0.02	\$0.02
DeepSeek-V3	2480	1600	\$0.01	\$0.01
GPT-o1	2550	1650	\$0.45	\$0.55

7.7 Efficiency Results

Table 12 reports latency measurements. GPT-o1 exhibits significantly higher latency due to its reasoning-intensive architecture, while Llama3-70B provides the fastest response times. These tradeoffs must be considered in time-sensitive incident response scenarios.

Table 12: Latency measurements (seconds per task)

Model	Mean	Median	p95	Execution Time
GPT-4	12.4	11.8	18.6	8.3
GPT-4o	10.8	10.2	16.4	7.1
Claude-3.5-Sonnet	11.2	10.6	17.1	7.4
Llama3-70B	8.6	8.1	13.2	5.6
DeepSeek-V3	9.3	8.8	14.0	6.1
GPT-o1	18.7	17.9	28.5	12.3

7.8 Platform Validation

Table 13 reports results on genuine platforms. The system demonstrates consistent performance across different platforms, with XuanJi showing the highest success rate.

Table 13: Platform validation results (successes/total cases)

Platform	Successes	Total Cases	Success Rate
TryHackMe	7	9	77.8%
XuanJi	5	6	83.3%
ZGSF	4	6	66.7%

7.9 Failure Analysis

7.9.1 Error Taxonomy

Table 14: Formal error taxonomy for failure analysis

Category	Definition	Examples
Strategic Formulation	Incorrect investigation strategy, hypothesis testing, or resource prioritization	Wrong investigation path, inefficient evidence collection, inappropriate containment strategy
Command Generation	Syntactically or semantically incorrect commands for target systems	Invalid syntax, wrong command options, system-incompatible commands
Evidence Omission	Missing critical evidence or contextual insight required for accurate analysis	Overlooking log entries, missing network indicators, ignoring system artifacts
Context Loss	Loss of information across phases of investigation, requiring re-establishment of context	Forgetting earlier findings in multi-step investigation, context window overflow
Hallucination	Generation of information not supported by evidence or system state	Fabricated log entries, incorrect system states, imaginary commands
Privacy Violation	Exposure of sensitive information in outputs without authorization	PII disclosure, credential exposure, proprietary information leakage

7.9.2 Coding Rules

- Error categories are mutually exclusive; each failed task is assigned exactly one primary category

Figure 2: Distribution of failure causes across evaluated incident response tasks (n=84 failed tasks)

- Secondary categories may be recorded but primary category is the root cause
- Multiple independent reviewers (2) with disagreement resolution through consensus

7.9.3 Failure Distribution

The failure distribution reveals:

- Strategic formulation errors: 44% (most common)
- Command generation errors: 19%
- Evidence omission errors: 18%
- Context loss errors: 13%
- Hallucination and privacy: 6%

7.10 Privacy Evaluation

7.10.1 Sensitive Data Taxonomy

Sensitive data categories are defined as follows:

- **PII:** Usernames, email addresses, user IDs, personal information
- **Credentials:** Passwords, API keys, authentication tokens
- **Internal addresses:** IP addresses, hostnames, internal network information
- **System logs:** System states, configurations, installed software
- **Incident data:** Attack details, vulnerability information, response actions

7.10.2 Privacy Evaluation Protocol

The privacy evaluation protocol consists of:

1. **Data injection:** Synthetic sensitive data is introduced into forensic artifacts
2. **Processing:** System processes artifacts through normal workflow
3. **Output inspection:** All outputs are inspected for sensitive data exposure
4. **Comparison:** Exposure rates compared between baseline and proposed system

Table 15: Privacy leakage comparison

Sensitive Data Type	Baseline Leakage	System Leakage
PII	18.5%	3.7%
Credentials	22.2%	5.6%
Internal Addresses	14.8%	1.9%
System Logs	11.1%	1.9%
Incident Data	16.7%	3.7%
Overall	16.7%	3.3%

7.10.3 Privacy Results

8. Discussion

8.1 Knowledge Cutoff Analysis

Models with more recent knowledge cutoffs generally performed better on recent attack techniques. However, this correlation requires formal statistical analysis; the observed trend is reported as preliminary observation rather than established correlation. This observation aligns with recent findings on AI model evaluation methodologies [23]. Code reasoning capabilities in LLMs have been shown to correlate with task performance in security-relevant applications [22].

8.2 Open-Source vs. Commercial Models

Open-source models (Llama3-70B, DeepSeek-V3) provided competitive performance at significantly lower cost, suggesting viability for budget-constrained deployments. Commercial models (GPT-4, Claude-3.5-Sonnet) offered advantages in reasoning complexity and API stability. The cost-performance tradeoff is particularly important for organizations with limited security budgets [18]. The broader implications of AI augmentation on employment patterns in security operations have been explored in recent work [21].

8.3 Practical Implications for SOC Operations

The findings suggest the following practical implications for security operations centers. First, multi-agent systems can effectively automate incident response planning and execution, reducing the cognitive burden on human analysts while maintaining accuracy. The decomposition of IR workflows into specialized agent roles mirrors the organizational structure of professional SOC teams, enabling parallel processing of complex investigations.

Second, privacy controls must be embedded by design rather than retrofitted into existing systems. The architectural controls implemented in our framework data minimization, local processing, and reflective privacy checks demonstrate measurable reductions in sensitive data leakage. Organizations deploying LLM-based security systems should prioritize these design patterns to maintain compliance with data protection regulations. Responsible AI practices in security contexts require careful attention to privacy implications [2]. The integration of ethical considerations into AI-driven security systems has been highlighted as a critical success factor [11].

Third, cost-performance tradeoffs necessitate careful model selection based on operational requirements. Open-source models offer compelling economics for budget-constrained deployments, while commercial models provide advantages in reasoning complexity and API stability. Organizations should

evaluate these tradeoffs against their specific incident response volumes, latency requirements, and available resources.

Fourth, human oversight remains essential for safety-critical decisions despite automation capabilities. The human-in-the-loop policy established in this work provides a template for balancing autonomous operation with human judgment, particularly for high-stakes actions such as production system modifications and data exfiltration prevention. This balanced approach aligns with recommendations for responsible AI implementation in security-critical environments [6].

9. Limitations and Threats to Validity

9.1 Internal Validity

- Variability in LLM behavior may affect reproducibility
- Prompt sensitivity requires careful configuration
- Manual failure classification introduces potential subjectivity

9.2 External Validity

- Results are based on cyber-range scenarios, not production deployments
- Performance may vary with different security environments and incident types
- Attack patterns evolve; benchmark may not represent all threat scenarios

9.3 Construct Validity

- Metrics may not fully capture incident response quality
- Privacy leakage measurements may not reflect real-world adversary capabilities
- Success criteria may be overly strict for realistic operations

9.4 Reproducibility Validity

- Models evolve; results may change with different API versions
- Randomness in LLM responses requires careful seed management
- Prompt sets require complete documentation for replication

10. Conclusion and Future Work

This paper presents a privacy-aware multi-agent LLM framework for cyber incident response, evaluated on a comprehensive benchmark of realistic incident-response scenarios. The key findings demonstrate that the multi-agent architecture achieves subtask completion rates of 72.3–86.9% across LLM backbones, outperforming baseline models by 13.9–21.5 percentage points. Strategic formulation errors account for 44% of failures, representing the primary limitation of current LLM capabilities, followed by command generation (19%) and evidence omission (18%). Privacy leakage is reduced from 16.7% to 3.3% through architectural controls, demonstrating the effectiveness of embedded privacy mechanisms.

Open-source models offer competitive performance at significantly lower cost, making them viable for budget-constrained deployments.

The contributions of this work include a reproducible IR benchmark, a modular multi-agent architecture with four specialized components, quantitative privacy evaluation, and comprehensive empirical assessment across six LLM variants. The system's ability to maintain context across extended investigations while generating accurate commands for system execution represents a significant advancement over existing LLM applications in cybersecurity, which remain limited in scope.

10.1 Future Work

Future research should focus on the following directions. First, reasoning improvements are needed to address strategic formulation errors through enhanced planning and hypothesis generation capabilities. This could involve incorporating external knowledge bases and structured reasoning frameworks. Second, privacy preservation techniques should be expanded to implement and evaluate federated learning, differential privacy, and secure multi-party computation methods for LLM-based security systems, enabling collaborative learning while maintaining data confidentiality. Third, human-AI collaboration models warrant investigation to establish effective human oversight strategies that balance automation benefits with human judgment, particularly for high-stakes decisions. Fourth, knowledge integration approaches should incorporate external threat intelligence sources and security databases beyond LLM internal knowledge to improve accuracy on emerging threats. Fifth, deployment validation in production SOC environments is essential to evaluate system performance under real operational conditions, including integration with existing security infrastructure and analyst workflows. Finally, explainability and transparency improvements will help secure stakeholder trust and validate reliability in critical security operations.

References

- [1] A. Agarwal. Adaptive security orchestration: Intelligent policy enforcement. In *2025 World Skills Conference on Universal Data Analytics and Sciences (WorldSUAS)*, pages 1–6, 2025.
- [2] H. Kasireddy. Responsible ai in fintech interfaces: Balancing innovation with ethical user experience design. *Journal of Data Analysis and Artificial Intelligence*, 4(3), 2025.
- [3] I. H. Sarker, A. S. M. Kayes, S. Badsha, H. Alqahtani, P. Watters, and A. Ng. Cybersecurity data science: An overview from machine learning perspective. *Journal of Big Data*, 7(1):41, 2020.
- [4] P. Dixit and S. Silakari. Deep learning algorithms for cybersecurity applications: A technological and status review. *Computer Science Review*, 39:100317, 2021.
- [5] Jasmin Bharadiya. Machine learning in cybersecurity: Techniques and challenges. *European Journal of Technology*, 7(2):1–14, 2023.
- [6] Raminder Kaur, Dušan Gabrijelčič, and Tomaž Klobučar. Artificial intelligence for cybersecurity: Literature review and future research directions. *Information Fusion*, 97:101804, 2023.
- [7] Humza Naveed, Asad Ullah Khan, Shi Qiu, Muhammad Saqib, Saeed Anwar, Muhammad Usman, ..., and Ajmal Mian. A comprehensive overview of large language models. *ACM Transactions on Intelligent Systems and Technology*, 16(5):1–72, 2025.

- [8] Jean Kaddour, Joshua Harris, Maximilian Mozes, Herbie Bradley, Roberta Raileanu, and Robert McHardy. Challenges and applications of large language models. *arXiv preprint arXiv:2307.10169*, 2023.
- [9] R. Sugumar. Next-generation security operations center (soc) resilience: Autonomous detection and adaptive incident response using cognitive ai agents. *International Journal of Technology, Management and Humanities*, 10(2):62–76, 2024.
- [10] Yupeng Chang, Xu Wang, Jindong Wang, Yuan Wu, Linyi Yang, Kaijie Zhu, ..., and Xing Xie. A survey on evaluation of large language models. *ACM Transactions on Intelligent Systems and Technology*, 15(3):1–45, 2024.
- [11] Neil Richards. *Why privacy matters*. Oxford University Press, 2022.
- [12] Daniel J. Solove. The limitations of privacy rights. *Notre Dame L. Rev.*, 98:975, 2022.
- [13] Daniel Schlette, Marco Caselli, and Günther Pernul. A comparative study on cyber threat intelligence: The security incident response perspective. *IEEE Communications Surveys & Tutorials*, 23(4): 2525–2556, 2021.
- [14] Atif Ahmad, Sean B. Maynard, Kevin C. Desouza, John Kotsias, M. T. Whitty, and Richard L. Baskerville. How can organizations develop situation awareness for incident response: A case study of management practice. *Computers & Security*, 101:102122, 2021.
- [15] S. P. Panda. Securing 5g critical interfaces: A zero trust approach for next-generation network resilience. In *2025 12th International Conference on Information Technology (ICIT)*, pages 141–146, 2025.
- [16] S. Malipeddi. Analyzing advanced persistent threats (apts) using passive honeypot sensors and self-organizing maps. In *2025 International Conference on Emerging Smart Computing and Informatics (ESCI)*, pages 1–7, 2025.
- [17] S. Parekh. Java platform security: A behavioral model for speculative execution vulnerabilities. In *2025 International Conference on Intelligent Communication Networks and Computational Techniques (ICICNCT)*, pages 1–7, 2025.
- [18] K. Bhatt. Risk-based optimization strategies for secure and efficient digital payment systems. In *2025 International Conference on Information, Implementation, and Innovation in Technology (I2ITCON)*, pages 1–6, 2025.
- [19] S. Sukhatankar. Beyond latent patterns: Reinterpreting ai model capabilities. TechRxiv preprint, 2025.
- [20] N. G. Camacho. The role of ai in cybersecurity: Addressing threats in the digital age. *Journal of Artificial Intelligence General Science (JAIGS) ISSN: 3006-4023*, 3(1):143–154, 2024.
- [21] F. Ted Tschang and Esteve Almirall. Artificial intelligence as augmenting automation: Implications for employment. *Academy of Management Perspectives*, 35(4):642–659, 2021.
- [22] Alex Gu, Baptiste Rozière, Hugh Leather, Armando Solar-Lezama, Gabriel Synnaeve, and S. I. Wang. Cruxeval: A benchmark for code reasoning, understanding and execution. *arXiv preprint arXiv:2401.03065*, 2024.

- [23] Wanjun Zhong, Ruixiang Cui, Yiduo Guo, Yaobo Liang, Shuai Lu, Yanlin Wang, ..., and Nan Duan. Agieval: A human-centric benchmark for evaluating foundation models. In *Findings of the Association for Computational Linguistics: NAACL 2024*, pages 2299–2314, June 2024.
- [24] Serkan Savaş and Sibel Karataş. Cyber governance studies in ensuring cybersecurity: an overview of cybersecurity governance. *International Cybersecurity Law Review*, 3(1):7–34, 2022.
- [25] U. M. Mbanaso, L. Abrahams, and K. C. Okafor. *Research Techniques for Computer Science, Information Systems and Cybersecurity*. Springer, 2023.