POLAR: LLM-POWERED SYNTHESIS OF REAL-WORLD CYBER EVIDENCE FOR PRIORITIZATION AND MITIGATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Cyber threat analysis increasingly depends on evidence distributed across vendor advisories, vulnerability databases, and threat intelligence sources. Turning these fragmented observations into timely decisions requires models to connect technical severity with evolving exploitation evidence and available defensive actions. We present POLAR, an LLM-powered framework for synthesizing real-world cyber evidence into threat-centric assessments for prioritization and mitigation. POLAR first disentangles overlapping incidents and grounds each threat in source-linked evidence. For prioritization, it infers severity metrics from cyber evidence and combines the resulting assessment with temporally ordered exploitation signals to estimate near-term exploitation likelihood. For mitigation, it links the synthesized threat data to authoritative remediation knowledge and organizes applicable actions according to threat urgency and operational constraints. We evaluate POLAR on real-world vulnerability evidence collected from public resources and compare it with multiple baselines. Across heterogeneous incidents and zero-day settings, POLAR improves threat ranking and mitigation retrieval while producing evidence-linked intermediate assessments that support analyst inspection. The results establish evidence synthesis as a practical foundation for LLM-based cyber decision support across related security tasks.

1 INTRODUCTION

Cyber threat analysis increasingly depends on evidence that is fragmented across real-world security sources. A vulnerability may first appear in a vendor disclosure, later receive structured metadata in a vulnerability database, and then acquire new operational significance as exploitation evidence emerges. We use the term **real-world cyber evidence** to describe these observations and authoritative records as they accumulate around a threat. The central analytical problem is to turn this evolving evidence into decisions about threat urgency and remediation. Standard mechanisms such as the Common Vulnerability Scoring System (CVSS) (FIRST, 2019a;b) and the Exploit Prediction Scoring System (EPSS) (Jacobs et al., 2021b; 2019) provide useful views of severity and exploitation likelihood, yet threat analysis still requires evidence to be synthesized across sources and interpreted in context.

Existing cyber analysis systems capture only part of this process. Rule-based tools and platforms are effective when signatures and decision rules are known (White et al., 2013; Sancho et al., 2020) . Their coverage weakens as threats and evidence evolve (Barr-Smith et al., 2021; Dong et al., 2023; Liu et al., 2024). Learning-based systems offer greater flexibility, including vulnerability classifiers (Almutairi, 2026; Zhan et al., 2021) and log-analysis models (Sasikrishna et al., 2026; Pakshad et al., 2026), though they commonly rely on fixed input representations and remain sensitive to shifts in CTI sources. A deeper limitation is that threat signals are often processed as isolated features. In practice, their meaning changes with context (Jørgensen & Ma, 2026). A moderate-severity vulnerability can become urgent after credible exploitation appears, while a high-severity vulnerability may pose limited immediate risk under restrictive exploitation conditions (Eling et al., 2023). Reliable analysis therefore depends on connecting evidence across technical and temporal context.

Large language models (LLMs) offer opportunities for effective threat analysis given their capability to synthesize evidence expressed at different levels of abstraction (Nguyen et al., 2026), such as an

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

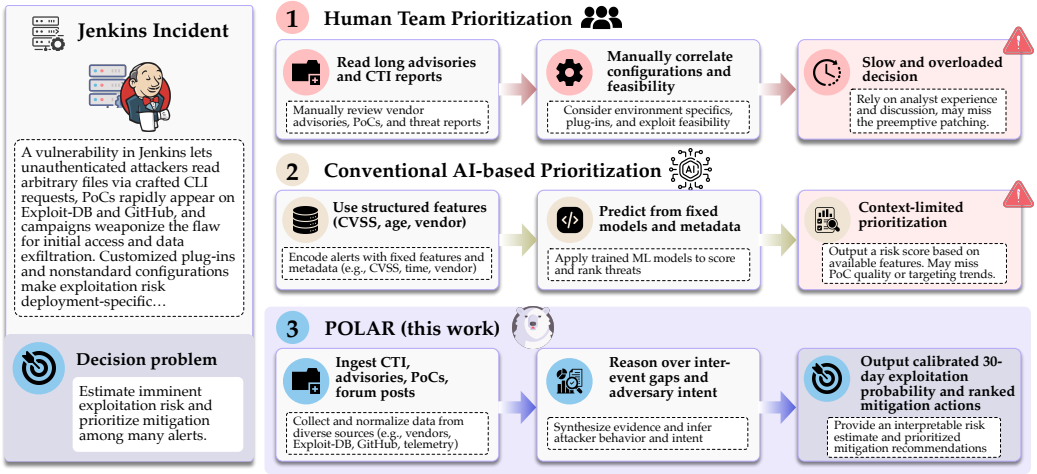

Figure 1: A motivating example of different threat-prioritization workflows, wherein POLAR synthesizes heterogeneous cyber evidence to support timely assessment and mitigation planning.

authentication requirement in an advisory and a subsequent report of active exploitation. Recent work has applied LLMs to penetration testing (Deng et al., 2024), attack-path analysis (Prapty et al., 2024; Bahlibi et al., 2026; Meng et al., 2025b), and security-event reasoning (Uetz et al., 2024; Meng et al., 2025a). The remaining challenge is to organize this evidence before it is used for a decision. However, direct prompt-based agent design can conflate incidents or obscure the evidence that supports a conclusion (Ramachandran, 2026). This leads to our research question: How can LLMs synthesize heterogeneous real-world cyber evidence for structured assessments and downstream decisions?

This Work. We present POLAR (Figure 1), an LLM-powered framework that builds a threat-centric evidence profile from unstructured incidents and authoritative external sources. POLAR separates overlapping threat descriptions, grounds each threat in source-linked metadata, and preserves evidence relevant to later analysis. This profile serves as the common input to two application scenarios studied in this work: **(I) Prioritization:** POLAR combines two complementary assessments. It derives CVSS metrics from evidence specific to each severity dimension, then applies the official scoring procedure to obtain a standardized severity estimate. It also organizes exploitation evidence by time and uses the resulting history to estimate near-term exploitation likelihood. The combined view captures the consequence of successful exploitation together with the current evidence that exploitation may occur. **(II) Mitigation:** POLAR connects the same threat profile to authoritative remediation knowledge and identifies actions applicable to the affected product and exploitation context. The resulting recommendations use the threat assessment and operational constraints to connect evidence of urgency with evidence for action.

We evaluate POLAR on real-world vulnerability data enriched with temporal threat events, covering a broad range of CVEs and cross-referencing evidence from Exploit-DB (Offensive Security, 2025), CISA KEV (Cybersecurity and Infrastructure Security Agency (CISA), 2024), VirusTotal (VirusTotal, 2004), and vendor advisories. Across comparisons with different baseline methods, POLAR consistently improves threat prioritization and remains stable under entangled and heterogeneous incident settings. The gains are significant for high-severity threats with active or rapidly evolving exploitation evidence, where POLAR produces more robust and timely prioritization. These results demonstrate the value of synthesizing heterogeneous real-world cyber evidence for decision support[1].

Overall, this work makes several contributions: (1) We formulate real-world cyber evidence synthesis as a unified foundation for LLM-based prioritization and mitigation. (2) We develop POLAR to organize heterogeneous cyber evidence into source-grounded threat assessments combining severity and exploitation dynamics. (3) We evaluate POLAR across prioritization and mitigation tasks, showing consistent gains over multiple baseline methods.

[1]We release our code at `https://anonymous.4open.science/r/LLM-Prioritization-01D2/`.

2 MOTIVATING EXAMPLE

Figure 1 uses a Jenkins vulnerability to illustrate the practical difficulty of threat prioritization. The vulnerability allows an unauthenticated attacker to read arbitrary files through crafted CLI requests. Soon after disclosure, public PoCs appear on Exploit-DB and GitHub, and attacker campaigns begin using the flaw for initial access and data exfiltration. The risk also varies across deployments because Jenkins instances may use different plug-ins and nonstandard configurations. A security team therefore needs to answer two questions quickly: How likely is this vulnerability to be exploited in the near term, and what should be mitigated first?

Human team prioritization: An analyst would first read the vendor advisory and CTI reports, then check the available PoCs and compare them with the local Jenkins configuration. The analyst may also need to determine whether the exposed CLI path is reachable, whether installed plug-ins change the attack surface, and whether the observed campaign applies to the local environment. This process can produce a careful assessment, but it takes time. When hundreds of alerts arrive together, analysts must repeat the same cross-checking for many threats while new exploitation evidence continues to appear. The resulting delay can consume the short window in which preemptive patching is most useful.

Conventional AI-based prioritization: A conventional model can score the same vulnerability much faster from structured inputs such as CVSS metrics, vulnerability age, vendor information, and available enrichment metadata. This works well when the relevant signals are already encoded as features. The difficulty is that some of the most useful signals arrive as text or emerge after disclosure. The quality of a new PoC, evidence of active targeting, changes in attacker behavior, or deployment-specific constraints may never appear in the feature vector. Missing patch or exploit metadata can further weaken the score. The model can therefore rank the vulnerability quickly while still missing changes that make the threat more urgent in practice.

POLAR prioritization: POLAR processes the same incident together with the surrounding cyber evidence. It collects the advisory, CTI reports, public PoCs, exploitation reports, and other source-linked observations, then organizes them into a temporal sequence such as disclosure $\rightarrow$ PoC $\rightarrow$ KEV $\rightarrow$ in-the-wild exploitation. The model examines how quickly these events occur, whether the PoC is readily weaponizable, and whether current attacker activity matches the affected environment. From this evidence, POLAR estimates the probability of exploitation within the next 30 days and combines that assessment with the threat's technical severity. It then links the prioritized threat to applicable mitigation evidence, such as vendor patches, workarounds, and defensive controls.

This example illustrates the key design principle of POLAR: cyber prioritization depends on evidence that changes over time and arrives in different forms. A useful system must keep those signals connected to the threat being assessed and carry them through to the mitigation decision.

3 POLAR: REAL-WORLD CYBER EVIDENCE SYNTHESIS

POLAR synthesizes heterogeneous cyber reports into threat-centric evidence for downstream security analysis. Figure 2 illustrates the workflow. Below we detail the design of POLAR.

3.1 REAL-WORLD CYBER EVIDENCE SYNTHESIS

Cyber evidence is distributed across reports that differ in scope, structure, and timing. A vendor advisory may describe several vulnerabilities together, while exploit repositories and later threat reports contribute new information about weaponization and deployment conditions. We formulate evidence synthesis as the construction of a structured representation (Algorithm 1) centered on an individual threat while retaining the provenance and temporal context of its supporting evidence.

Algorithm 1: Cyber Evidence Synthesis

Input: Incidents $\mathcal{I} = \{i_j\}_{j=1}^{N}$; evidence sources $\mathcal{D}$
Output: Evidence $\mathcal{Z} = \{z_k\}_{k=1}^{M}$
// Disentangle threat instances
foreach $i_j \in \mathcal{I}$ **do**
  $\mathcal{T}_j \leftarrow \phi_{\text{LLM}}(i_j) = \{\tau_{j,1}, \ldots, \tau_{j,L_j}\}$;
end
// Ground threats in external evidence
foreach $\tau_k \in \bigcup_j \mathcal{T}_j$ **do**
  $\mathcal{E}_k \leftarrow \psi(\tau_k, \mathcal{D})$;
  $\mathbf{m}_k \leftarrow \text{METADATA}(\tau_k, \mathcal{D})$;
  $z_k \leftarrow (\tau_k, \mathbf{m}_k, \mathcal{E}_k)$;
end
return $\mathcal{Z} = \{z_k\}_{k=1}^{M}$;

❶ **Threat disentanglement.** Let $\mathcal{I} = \{i_1, \ldots, i_N\}$ denote the raw incidents available at analysis time t. An

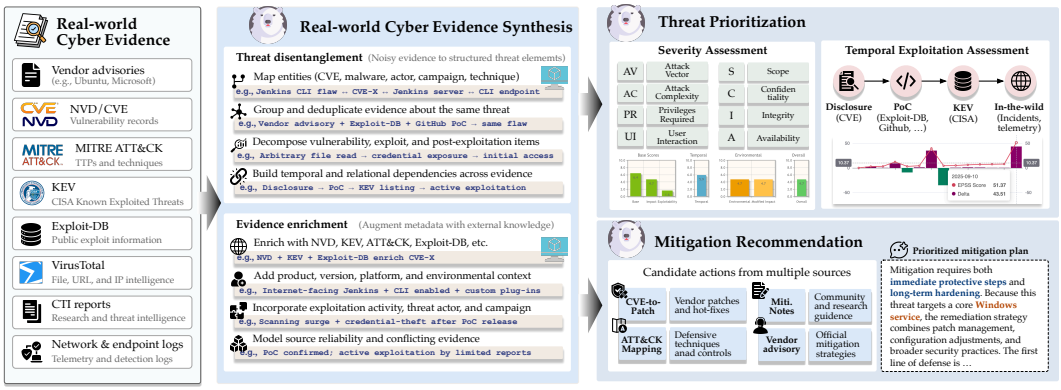

Figure 2: Overview of POLAR. Real-world cyber evidence is synthesized into structured threat profiles that support downstream threat prioritization and mitigation recommendation.

incident may describe several vulnerabilities, campaigns, or related indicators. POLAR applies an LLM-based separation operator ϕ_{LLM} to obtain canonical threat instances τ_k, while preserving the affected component, exploitation path, and deployment conditions associated with each instance. This separation reduces evidence conflation when multiple threats appear in the same report. Detailed instructions are provided in Appendix D.1.

❷ **Source-aware evidence enrichment.** For each threat τ_k, POLAR retrieves complementary evidence from external sources $\mathcal{D}$, including vulnerability databases, ATT&CK knowledge, exploitation feeds, and vendor or government advisories. The resulting set $\mathcal{E}_k = \psi(\tau_k, \mathcal{D}) = \{e_{k,1}, \ldots, e_{k,R_k}\}$ contains evidence units $e_{k,r} = (s_{k,r}, u_{k,r}, t_{k,r}, c_{k,r})$, where $s_{k,r}$ denotes the supporting content, $u_{k,r}$ as its source, $t_{k,r}$ as its timestamp when available, and $c_{k,r}$ as its analytical role. In parallel, POLAR extracts standardized metadata $\mathbf{m}_k$, including the CVE identifier, affected product, disclosure date, and ATT&CK mappings. Together they form the threat-centric evidence $z_k = (\tau_k, \mathbf{m}_k, \mathcal{E}_k)$.

Note that z_k includes severity and temporal evidence, thus facilitating subsequent CVSS assessment and exploitation analysis. Given an analytical query q (i.e., a task), POLAR selects the corresponding view $\mathcal{E}_k^{(q)} \subseteq \mathcal{E}_k$ for task-specific analysis while retaining provenance and contextual dependencies.

Example 1: Synthesizing evidence for a Print Spooler threat

Raw incident i_j: A Windows 11 report states that an authenticated remote attacker can abuse the Print Spooler through a crafted DCERPC request to load a DLL and execute code as NT AUTHORITY.

Threat instance τ_k: POLAR disentangles this report into a canonical Print Spooler threat, preserving its affected component, remote exploitation path, and privilege conditions.

Evidence profile z_k: POLAR extracts metadata $\mathbf{m}_k$ (e.g., the mapped CVE, affected versions, and ATT&CK techniques), and constructs $\mathcal{E}_k$ from source-linked evidence units $e_{k,r} = (s_{k,r}, u_{k,r}, t_{k,r}, c_{k,r})$. For example, "authenticated remote attacker" is retained as severity evidence, while a CISA KEV record contributes temporally grounded exploitation evidence.

Applying this procedure to all incidents yields $\mathcal{Z} = \{z_1, \ldots, z_M\}$, which is shared by the two applications below.

3.2 APPLICATION I: CYBER THREAT PRIORITIZATION

Threat prioritization combines two complementary assessments. The severity score s_k captures the technical consequences and exploitation conditions of a vulnerability, while the exploitation probability p_k reflects its evolving likelihood of real-world exploitation. Keeping both quantities explicit allows threats with similar static severity to be distinguished according to current threat dynamics. Algorithm 2 summarizes how POLAR derives these two assessments from z_k and combines them into the final priority ordering.

Algorithm 2: Threat Prioritization

Input: Evidence set $\mathcal{Z}$; CVSS guidance D_{CVSS}
Output: Priority ordering π_{pri}
foreach $z_k \in \mathcal{Z}$ **do**
  foreach *CVSS metric* i **do**
    $\mathcal{E}_k^{(i)} \leftarrow \text{SELECT}(\mathcal{E}_k, i)$;
    $v_k^{(i)} \leftarrow f_{\text{LLM}}^{(i)}(\mathcal{E}_k^{(i)}, D_{\text{CVSS}})$;
  end
  $s_k \leftarrow \mathcal{S}_{\text{CVSS}}(\mathbf{v}_k)$;
  $\mathcal{E}_k^{\text{temp}} \leftarrow \text{TEMPORALVIEW}(\mathcal{E}_k)$;
  $p_k \leftarrow f_{\text{LLM}}^{\text{exp}}(z_k, \mathbf{v}_k, \mathcal{E}_k^{\text{temp}})$;
  $r_k \leftarrow (z_k, s_k, p_k)$;
end
$\pi_{\text{pri}} \leftarrow \text{RANK}(\{r_k\}_{k=1}^{M})$;
return π_{pri};

❶ Evidence-grounded severity assessment. We represent technical severity through the CVSS vector $\mathbf{v}_k = (v_k^{(1)}, \ldots, v_k^{(d)}) \in \mathcal{Y}^{(1)} \times \cdots \times \mathcal{Y}^{(d)}$, where $\mathcal{Y}^{(i)}$ is the valid categorical domain of metric i. For each metric, POLAR selects a source-linked evidence view $\mathcal{E}_k^{(i)} \subseteq \mathcal{E}_k$ and predicts $v_k^{(i)} = f_{\mathrm{LLM}}^{(i)}(\mathcal{E}_k^{(i)}, D_{\mathrm{CVSS}})$ under the official CVSS instructions (FIRST, 2019a;b). Exposure evidence informs Attack Vector, exploitation preconditions inform Attack Complexity, authentication requirements inform Privileges Required, and victim-action dependencies inform User Interaction. Confidentiality, integrity, and availability are inferred from evidence describing the consequences of successful exploitation. Detailed metric-specific workflows are provided in Appendix D.2.1.

Once all categorical metrics are determined, the official scoring function computes $s_k = \mathcal{S}_{\mathrm{CVSS}}(\mathbf{v}_k)$, with $s_k \in [0, 10]$. The LLM thus interprets the source-linked evidence required by each CVSS dimension, while the standardized scoring procedure performs numerical aggregation.

Example 2: Evidence-grounded Attack Vector assessment

For the threat in Example 1, POLAR selects the source spans "`authenticated remote attacker`" and "`crafted DCERPC request`", together with protocol evidence indicating RPC over SMB/DCERPC. When local and remote cues coexist, the model checks which access mode is supported by the documented exploitation path. The resulting evidence supports **AV=Network**. The remaining CVSS metrics are assessed independently before the official scoring procedure computes s_k.

❷ Temporal exploitation assessment. Static severity is complemented by a temporally ordered evidence view $\mathcal{E}_k^{\mathrm{temp}} = \{(t_\ell, e_\ell, u_\ell)\}_{\ell=1}^{L_k}$, where each tuple records an exploitation-related observation, its timestamp, and its source. Such observations include vulnerability disclosure, public PoC release, Exploit-DB publication, CISA KEV listing, malware observations, and reports of exploitation in the wild. POLAR orders these events chronologically and preserves their temporal gaps, yielding trajectories such as disclosure $\rightarrow$ PoC $\rightarrow$ KEV $\rightarrow$ in-the-wild exploitation.

Conditioned on the threat evidence z_k, its CVSS representation $\mathbf{v}_k$, and the temporal evidence $\mathcal{E}_k^{\mathrm{temp}}$, POLAR estimates $p_k = f_{\mathrm{LLM}}^{\mathrm{exp}}(z_k, \mathbf{v}_k, \mathcal{E}_k^{\mathrm{temp}})$, where $p_k \in [0, 1]$ denotes the probability of observed exploitation within the next 30 days, following the prediction horizon used by EPSS (Jacobs et al., 2021b;a). The complete set of exploitation sources is described in Appendix D.2.2.

❸ Joint prioritization. The resulting $r_k = (z_k, s_k, p_k)$ couples synthesized evidence with standardized severity and dynamic exploitation likelihood. Given concurrent threats, POLAR compares the resulting r_k to produce the priority ordering π_{pri}. The explicit intermediate quantities also expose the severity and exploitation evidence underlying the final ranking.

3.3 APPLICATION II: EVIDENCE-GROUNDED MITIGATION RECOMMENDATION

Mitigation translates the synthesized threat assessment into concrete defensive actions. Algorithm 3 summarizes this process from remediation retrieval to the final mitigation ordering. Given z_k, POLAR retrieves candidate mitigations from an authoritative remediation corpus $\mathcal{D}_{\mathrm{rem}}$ containing vendor security bulletins, NVD references, CISA KEV notes, and MITRE ATT&CK mitigations. This produces $\mathcal{A}_k = f_{\mathrm{LLM}}^{\mathrm{ret}}(z_k, \mathcal{D}_{\mathrm{rem}}) = \{a_{k,1}, \ldots, a_{k,J_k}\}$, with each action linked to its supporting source. We organize the retrieved evidence into the four categories used in our evaluation: CVE-to-patch mappings, ATT&CK-aligned mitigations, mitigation notes or workarounds, and vendor advisories.

Algorithm 3: Mitigation Synthesis

Input: Evidence set $\{r_k\}_{k=1}^{M}$; remediation corpus $\mathcal{D}_{\mathrm{rem}}$
Output: Mitigation plan Π_{mit}
foreach $r_k = (z_k, s_k, p_k)$ **do**
  `// Retrieve candidate actions`
  $\mathcal{A}_k \leftarrow f_{\mathrm{LLM}}^{\mathrm{ret}}(z_k, \mathcal{D}_{\mathrm{rem}})$;
  `// Normalize and verify applicability`
  $\mathcal{A}_k \leftarrow$ NORMALIZE$(\mathcal{A}_k)$;
  $\mathcal{A}_k \leftarrow$ FILTERAPPLICABLE$(\mathcal{A}_k, z_k)$;
end
`// Synthesize ordered mitigation actions`
$\Pi_{\mathrm{mit}} \leftarrow f_{\mathrm{LLM}}^{\mathrm{mit}}(\{(r_k, \mathcal{A}_k)\}_{k=1}^{M})$;
return Π_{mit};

❶ Applicability grounding. Candidate actions are matched against the synthesized threat context before recommendation. Product and version metadata determine patch applicability, exploitation-path evidence identifies relevant compensating controls, and temporal evidence distinguishes current guidance from superseded remediation. POLAR also normalizes semantically overlapping actions while retaining their supporting provenance, producing a threat-specific action set $\mathcal{A}_k$.

❷ Mitigation ordering. Given the prioritized $\{r_k\}$ and associated candidate actions $\{\mathcal{A}_k\}$, POLAR produces $\Pi_{\mathrm{mit}} = f_{\mathrm{LLM}}^{\mathrm{mit}}(\{(r_k, \mathcal{A}_k)\}_{k=1}^{M})$. The ordering considers threat severity and exploitation like-

lihood together with mitigation-specific factors such as patch availability, workaround effectiveness, implementation complexity, and operational disruption. The synthesized evidence therefore connects the evidence establishing threat urgency with the evidence supporting an appropriate defensive response.

Output. POLAR returns the priority ordering π_{pri} together with the source-linked mitigation plan Π_{mit}. Both applications operate on the same threat-centric evidence: prioritization maps evidence to urgency, while mitigation maps the resulting assessment to actionable defense.

4 EXPERIMENT

We perform extensive evaluations on POLAR, aiming to answer the following research questions: **RQ$_1$:** How effective is POLAR in threat prioritization and mitigation recommendation compared with existing baselines? **RQ$_2$:** How does each component of POLAR contribute to its overall performance? **RQ$_3$:** How does POLAR perform across individual pipeline stages? **RQ$_4$:** Can POLAR generalize to zero-day vulnerability analysis?

LLMs. As baseline and backbone of POLAR, we evaluate with five representative LLMs: Gemini-3.1-Pro (**GE3.1**), GPT-5 (**G5**), LLaMA-3-70B-Instruct (**L70B**), Claude Opus 4.1 (**C4.1**), and Qwen3-30B-A3B-Instruct-2507 (**Q3**). We also evaluate three LLM-based cybersecurity-specialist agents: Foundation-Sec-8B-Instruct (**FS**) (Singer, 2025), Lily-Cybersecurity-7B (**LY**) (Labs, 2024), and ZySec-7B (**ZY**) (ZySec AI, 2024).

Data. We collect raw threat incidents from high-fidelity vendor advisories (e.g., Microsoft, Cisco, Oracle), security reports (e.g., Mandiant (Mandiant, 2004)), public CTI aggregators (e.g., the KEV Catalog (Cybersecurity and Infrastructure Security Agency (CISA), 2024)), and exploit repositories (e.g., Exploit-DB (Offensive Security, 2025) and GitHub PoCs). Each incident is converted into structured evaluations covering four stages of POLAR: (i) triage and metadata enrichment grounded in authoritative repositories such as CVE/NVD (CVE Program, 2024) and MITRE ATT&CK (Strom et al., 2018); (ii) CVSS assessment validated against official NVD vectors; (iii) exploitation likelihood referenced from the FIRST.org EPSS dataset (Jacobs et al., 2021b); and (iv) mitigation strategies collected from NVD references, KEV mitigation notes, and vendor security updates. We standardize incidents into stage-specific tasks and use official repository assignments as reference answers when ambiguity arises. Dataset statistics and quality-control procedures are reported in Appendix E.1.

Baselines. We compare POLAR with representative non-LLM and LLM-based methods for vulnerability analysis and mitigation mapping. For threat prioritization, we include CVSS-BERT (Shahid & Debar, 2021), which predicts CVSS vectors from vulnerability descriptions using fine-tuned BERT classifiers, and DEEPCVA (Le et al., 2021), which jointly predicts CVSS metrics with multi-task deep learning. Both obtain severity scores using the official CVSS formula. For mitigation recommendation, we use CVE2CAPEC[2], a rule-based system that maps CVEs to CWEs and CAPEC attack patterns, and V2W-BERT (Das et al., 2021), which predicts CWE categories from CVE descriptions and retrieves associated mitigations from CWE definitions.

We also include two recent LLM-based cyber analysis frameworks as end-to-end baselines. AUTOCVSS (Sanvito et al., 2025) prompts LLMs to predict CVSS vectors and derives threat rankings from the resulting severity scores. TRIAGE (Høst et al., 2025) maps CVEs to MITRE ATT&CK techniques using rule-based prompting and in-context learning; we then retrieve the corresponding official ATT&CK mitigations for evaluation. Since these systems have their own complete pipelines, POLAR is evaluated against them directly rather than layered on top. CVSS- and EPSS-based rankings serve as reference signals in our evaluation and are therefore not treated as baselines.

4.1 MAIN RESULTS: THREAT PRIORITIZATION AND MITIGATION RECOMMENDATION (RQ$_1$)

Setting. We focus on the two most decision-critical outcomes in operational CTI workflows: (1) *threat prioritization*, evaluated by ranking-based metrics (NDCG@5 and Kendall's τ) against reference urgency order, and (2) *mitigation recommendation*, evaluated by the F1 score between generated mitigation items and database-sourced references. We compare POLAR against backbone general-

[2] https://github.com/Galeax/CVE2CAPEC

Table 1: Evaluation of threat prioritization and mitigation recommendation (mean±std across five runs). Ranking quality is measured by NDCG@5 and Kendall's τ, and mitigation retrieval is evaluated by F1 against reference mitigation items. "N/A" represents the baseline is not capable for corresponding tasks. All results are the mean and standard deviation over five runs. All metrics except Kendall's τ are reported in percentage (%).

Model		Prioritization Correctness		Mitigation Retrieval			
		NDCG@5	Kendall's τ	CVE-to-Patch	ATT&CK Mapping	Mitigation Note	Vendor Advisory
Non-LLM heuristic baseline							
	CVSS-BERT	51.2 ± 0.5	0.361 ± 0.012	N/A	N/A	N/A	N/A
	DEEPCVA	55.7 ± 0.4	0.404 ± 0.010	N/A	N/A	N/A	N/A
	CVE2CAPEC	N/A	N/A	29.8 ± 0.9	N/A	24.1 ± 1.5	26.8 ± 0.6
	V2W-BERT	N/A	N/A	33.6 ± 0.6	N/A	28.7 ± 1.1	30.9 ± 0.7
LLM-based cyber analysis framework							
	AUTOCVSS	59.8 ± 1.2	0.447 ± 0.007	N/A	N/A	N/A	N/A
	TRIAGE	N/A	N/A	N/A	54.2 ± 1.2	N/A	N/A
Cybersecurity-specialized agent							
POLAR +	**FS**	68.5 ± 0.8 (↑ 8.3)	0.553 ± 0.009 (↑ 0.101)	73.4 ± 1.3 (↑ 11.3)	63.5 ± 1.5 (↑ 13.2)	69.3 ± 1.4 (↑ 9.1)	62.5 ± 0.9 (↑ 7.3)
	LY	69.4 ± 0.7 (↑ 9.2)	0.564 ± 0.016 (↑ 0.112)	71.7 ± 0.8 (↑ 9.6)	62.4 ± 0.8 (↑ 12.1)	68.3 ± 0.7 (↑ 8.1)	61.6 ± 0.9 (↑ 6.4)
	ZY	70.5 ± 0.6 (↑ 10.3)	0.575 ± 0.016 (↑ 0.123)	74.2 ± 0.8 (↑ 12.1)	64.6 ± 1.3 (↑ 14.3)	70.4 ± 0.8 (↑ 10.2)	63.4 ± 1.4 (↑ 8.2)
General-purpose LLM							
POLAR +	**GE3.1**	85.8 ± 1.3 (↑ 24.9)	0.768 ± 0.017 (↑ 0.309)	92.7 ± 1.4 (↑ 30.1)	80.9 ± 0.5 (↑ 29.8)	89.1 ± 1.0 (↑ 28.8)	81.5 ± 1.0 (↑ 26.2)
	G5	87.3 ± 0.7 (↑ 27.1)	0.783 ± 0.019 (↑ 0.331)	94.3 ± 0.9 (↑ 32.2)	81.5 ± 1.1 (↑ 31.2)	90.3 ± 1.3 (↑ 30.1)	82.3 ± 0.7 (↑ 27.1)
	L70B	76.5 ± 0.5 (↑ 16.3)	0.673 ± 0.016 (↑ 0.221)	82.4 ± 1.1 (↑ 20.3)	74.4 ± 1.3 (↑ 24.1)	81.4 ± 1.0 (↑ 21.2)	73.5 ± 0.5 (↑ 18.3)
	C4.1	83.4 ± 0.7 (↑ 23.2)	0.743 ± 0.006 (↑ 0.291)	90.4 ± 0.8 (↑ 28.3)	80.4 ± 1.2 (↑ 30.1)	88.4 ± 1.0 (↑ 28.2)	80.5 ± 0.5 (↑ 25.3)
	Q3	80.6 ± 0.4 (↑ 20.4)	0.714 ± 0.014 (↑ 0.262)	87.2 ± 1.1 (↑ 25.1)	77.4 ± 0.8 (↑ 27.1)	86.3 ± 0.4 (↑ 26.1)	78.5 ± 0.4 (↑ 23.3)

purpose LLMs, cybersecurity-specialist agents, and representative non-LLM baselines. We also compare POLAR with recent LLM-based cyber analysis frameworks.

POLAR consistently outperforms non-LLM baselines on both prioritization and mitigation. Table 1 shows that severity-only predictors (e.g., CVSS-BERT, DEEPCVA) provide limited prioritization accuracy, as they rely solely on predicted CVSS scores without considering exploitation likelihood or contextual threat intelligence. In contrast, POLAR significantly improves both NDCG@5 and Kendall's τ by jointly reasoning over severity (s_k), exploitation likelihood (p_k), and enriched CTI evidence. For mitigation recommendation, heuristic approaches like CVE2CAPEC achieve low F1 scores due to their reliance on predefined linkage rules that fail to capture nuanced contextual relationships. Similarly, V2W-BERT is limited by its single-source knowledge dependency on CWE definitions, which often provide generic rather than actionable mitigations. POLAR consistently surpasses these baselines, particularly on mitigation notes and vendor advisories that require contextual reasoning beyond pattern matching.

4.2 ABLATION STUDY (RQ_2)

Setting. To understand the contribution of individual components in POLAR, we conduct an ablation study by selectively removing one stage from the pipeline at a time. We evaluate three representative backbone models: Gemini-3.1-Pro (**GE3.1**), GPT-5 (**G5**), and LLaMA-3-70B-Instruct (**L70B**).

For each backbone, we consider the following ablations: (i) *w/o Triage*, which removes incident disentanglement and metadata grounding; (ii) *w/o Static Analysis*, which bypasses structured CVSS reasoning; and (iii) *w/o Exploitation Analysis*, which excludes exploitation likelihood modeling. All variants are evaluated using the same metrics as in the main experiment.

All components contribute to the overall effectiveness of POLAR. Table 2 shows that removing any single stage leads to consistent degradation across both prioritization and mitigation metrics. Among the three components, triage removal causes the largest performance drop, particularly in prioritization accuracy, highlighting the importance of disentangling raw incidents and grounding them in authoritative metadata. Excluding static analysis primarily affects severity-sensitive ranking and patch alignment, while removing exploitation analysis has a more pronounced impact on mitigation ordering when multiple medium-severity threats compete. These results indicate that POLAR's performance gains arise from the joint interaction of its components.

4.3 BREAKDOWN ANALYSIS ACROSS PIPELINE STAGES (RQ_3)

We analyze POLAR by breaking down its performance across individual pipeline stages: CTI triage, static analysis, and exploitation forecasting. This analysis contextualizes the gains observed in the

Table 2: Ablation results on prioritization and mitigation (mean±std across five runs). Each block uses a different backbone model and reports the performance of the full POLAR pipeline and variants with one component removed. Results are reported as mean ± standard deviation over five runs. All metrics except Kendall's τ are reported in percentage (%).

Variant	Prioritization Correctness		Mitigation Retrieval			
	NDCG@5	Kendall's τ	CVE-to-Patch	ATT&CK Mapping	Mitigation Note	Vendor Advisory
Gemini-3.1-Pro (GE3.1)						
Full POLAR	85.8 ± 1.2	0.768 ± 0.014	92.7 ± 1.0	80.9 ± 1.0	89.1 ± 1.4	81.5 ± 1.0
w/o Triage	78.6 ± 1.4	0.679 ± 0.011	84.6 ± 0.6	73.4 ± 1.0	80.9 ± 0.7	74.2 ± 0.4
w/o Static Analysis	81.1 ± 1.2	0.704 ± 0.009	86.8 ± 1.2	75.5 ± 1.1	83.4 ± 0.6	76.8 ± 0.8
w/o Exploit. Analysis	82.4 ± 1.2	0.718 ± 0.009	87.7 ± 1.3	76.7 ± 0.8	84.9 ± 0.9	78.1 ± 1.2
GPT-5 (G5)						
Full POLAR	87.3 ± 0.9	0.783 ± 0.019	94.3 ± 1.6	81.5 ± 1.3	90.3 ± 0.9	82.3 ± 1.5
w/o Triage	80.1 ± 0.7	0.693 ± 0.013	86.2 ± 0.9	74.2 ± 1.2	82.1 ± 1.4	75.1 ± 1.3
w/o Static Analysis	82.4 ± 1.0	0.718 ± 0.014	88.3 ± 1.4	76.1 ± 1.4	84.6 ± 0.6	77.6 ± 1.1
w/o Exploit. Analysis	83.8 ± 0.6	0.732 ± 0.011	89.1 ± 1.1	77.4 ± 1.1	86.1 ± 0.7	78.9 ± 0.8
LLaMA-3-70B (L70B)						
Full POLAR	76.5 ± 0.7	0.673 ± 0.017	82.4 ± 0.5	74.4 ± 0.7	81.4 ± 0.5	73.5 ± 1.0
w/o Triage	73.2 ± 1.3	0.624 ± 0.008	78.1 ± 0.9	69.1 ± 1.1	75.4 ± 0.8	68.3 ± 1.4
w/o Static Analysis	75.4 ± 1.0	0.648 ± 0.008	80.2 ± 0.5	70.9 ± 1.6	77.6 ± 0.7	70.1 ± 0.5
w/o Exploit. Analysis	76.8 ± 0.6	0.662 ± 0.011	81.1 ± 1.1	72.1 ± 1.2	78.9 ± 0.6	71.4 ± 1.1

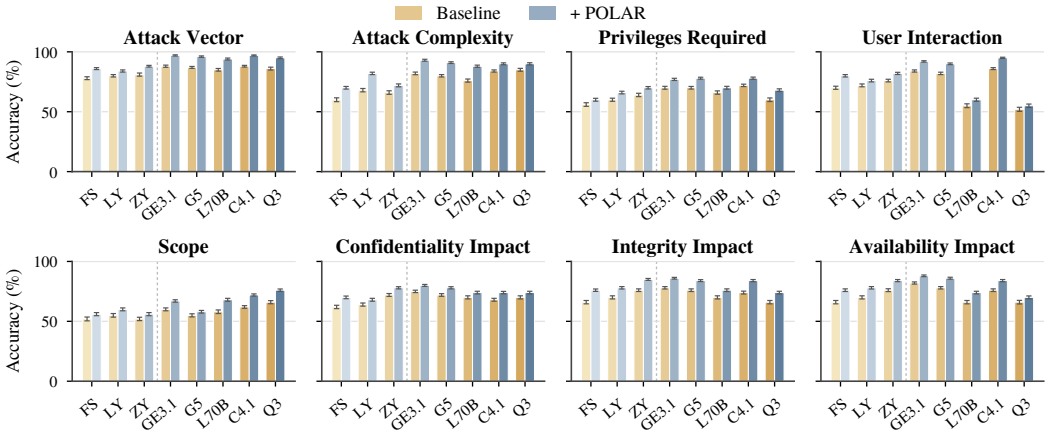

Figure 3: Static analysis performance (per-metric accuracy) across baseline models and POLAR.

main experiment (RQ_1) and ablation study (RQ_2), and highlights how each stage contributes to the overall effectiveness of the system.

CTI triage establishes a reliable foundation for downstream reasoning. Across evaluated LLM backbones, POLAR consistently improves incident disentanglement and metadata enrichment, especially for entangled advisories and knowledge-grounded attributes such as ATT&CK techniques and exploitation status. By converting unstructured CTI into structured, aligned representations, the triage stage reduces ambiguity in downstream reasoning and mitigates error propagation to later stages.

Static analysis benefits from evidence-grounded reasoning. Figure 3 shows that POLAR improves classification accuracy across all eight CVSS base metrics, with particularly notable gains on context-dependent dimensions such as *Attack Vector* and *Attack Complexity*. These improvements indicate that combining retrieval-grounded evidence from CTI triage with structured reasoning enables more reliable interpretation of vulnerability descriptions than severity-only prediction.

In addition to per-metric accuracy, POLAR also reduces overall severity scoring error, as reflected by consistently lower RMSE in Figure 4. This indicates that improvements are not limited to isolated CVSS fields, but translate into more reliable aggregate severity assessments that directly support prioritization decisions.

Exploitation analysis enhances robustness under dynamic threat conditions. For exploitation forecasting, POLAR shows improved robustness across different exploitation trends and historical

context windows. While baseline models can track gradual changes, POLAR better adapts to abrupt shifts by linking temporal signals to broader threat narratives, which is critical when new exploits or proof-of-concept releases rapidly alter risk profiles.

Overall, the breakdown analysis demonstrates that POLAR 's performance gains arise from the complementary interaction of its pipeline stages rather than improvements isolated to any single component. Detailed quantitative results, additional figures, and per-stage analyses are provided in Appendix E.2.1, Appendix E.2.4, and Appendix E.2.5.

4.4 ADDITIONAL EXPERIMENTS (RQ_4)

We further evaluate POLAR on zero-day vulnerabilities to assess its generalization beyond disclosed and well-documented threats. This setting focuses on vulnerabilities that lack complete metadata or official CVE annotations at the time of analysis, reflecting realistic early-stage threat assessment scenarios. Due to space limitations, we defer detailed analyses to the appendix: zero-day experiments (Appendix E.2.3), breakdown analysis with quantitative results for individual pipeline stages (Appendix E.2.1, E.2.4, E.2.5), and dataset statistics (Appendix E.1).

5 RELATED WORK

Threat prioritization. Traditional approaches to threat prioritization primarily relied on heuristic-driven models. Statistical and probabilistic methods, including Bayesian models and attack graphs, have been used to assess risk propagation and exploit likelihood within specific infrastructures (Ingols et al., 2009; Poolsappasit et al., 2012). Other approaches integrate expert knowledge with structured taxonomies, such as MITRE's ATT&CK framework, to map observed adversary techniques to potential system impacts (Strom et al., 2018). However, these methods generally emphasize static attributes and predefined rules, making them less effective in rapidly evolving threat environments.

LLMs for cybersecurity tasks. Surveys report broad LLM capability in code analysis, vuln detection/repair, and broader security workflows (Hasanov et al., 2024; Das et al., 2025; Zhou et al., 2025; Chen et al., 2025). Task-specific systems show gains in penetration testing via tool use and chain-of-thought orchestration (PentestGPT) (Deng et al., 2024), and in attack modeling via retrieval-augmented generation over threat knowledge bases (Prapty et al., 2024). Evaluation and deployment studies surface challenges around robustness, hallucination, and operational risk (McGraw et al., 2024; Keppler et al., 2024). Most prior efforts, however, target isolated tasks rather than end-to-end prioritization.

LLM agents and automation. LLM-as-agent paradigms combine reasoning, tools, and environment interaction for complex goals (Dong et al., 2023). In security, automated attack-graph construction and reasoning trace feasible paths and preconditions (Ou et al., 2005; Sheyner et al., 2002), while multi-agent designs specialize roles across assessment, exploitation prediction, and mitigation planning (Sancho et al., 2020). Retrieval-augmented agents and deep log-analysis models enhance detection under real-world data heterogeneity (Du et al., 2017; Zhang et al., 2022; Uetz et al., 2024).

6 CONCLUSION

We present POLAR, an LLM-powered framework for synthesizing real-world cyber evidence for threat prioritization and mitigation. POLAR organizes heterogeneous evidence into threat-centric representations, combines severity with temporal exploitation signals for prioritization, and connects the resulting assessments to source-linked mitigation actions. Our experiments show consistent improvements across prioritization and mitigation tasks, including heterogeneous and zero-day settings. These results demonstrate the value of evidence synthesis for practical LLM-based cyber decision support. **Ethics statement:** This work uses publicly available vulnerability and cyber threat intelligence sources and does not involve human subjects or private user data. POLAR is designed for defensive threat assessment and mitigation support, with its outputs intended to assist security analysts in reviewing and prioritizing evidence-based actions.

REFERENCES

Nawal Almutairi. Llm-powered proactive cyber-defense framework using cyber-threat indicators collected from x platform. *Electronics*, 15(6):1305, 2026.

Kibrom Bahlibi, Hoang Long Nguyen, Erdogan Dogdu, Roya Choupani, and William Mitchell. Beyond single-hop: Link prediction through multi-hop reasoning in malware knowledge graphs. In *2026 IEEE 5th International Conference on AI in Cybersecurity (ICAIC)*, pp. 1–10. IEEE, 2026.

Frederick Barr-Smith, Xabier Ugarte-Pedrero, Mariano Graziano, Riccardo Spolaor, and Ivan Martinovic. Survivalism: Systematic analysis of windows malware living-off-the-land. In *2021 IEEE Symposium on Security and Privacy (SP)*, pp. 1557–1574. IEEE, 2021.

Yuchen Chen, Weisong Sun, Chunrong Fang, Zhenpeng Chen, Yifei Ge, Tingxu Han, Quanjun Zhang, Yang Liu, Zhenyu Chen, and Baowen Xu. Security of language models for code: A systematic literature review. *ACM Transactions on Software Engineering and Methodology*, 2025.

Computer Emergency Response Team Coordination Center (CERT/CC). Cert security bulletins. `https://www.cert.org/`, 1988.

CVE Program. Common vulnerabilities and exposures (cve). `https://www.cve.org/`, 2024.

Cybersecurity and Infrastructure Security Agency (CISA). Known exploited vulnerabilities catalog. `https://www.cisa.gov/known-exploited-vulnerabilities-catalog`, 2024.

Badhan Chandra Das, M Hadi Amini, and Yanzhao Wu. Security and privacy challenges of large language models: A survey. *ACM Computing Surveys*, 57(6):1–39, 2025.

Siddhartha Shankar Das, Edoardo Serra, Mahantesh Halappanavar, Alex Pothen, and Ehab Al-Shaer. V2w-bert: A framework for effective hierarchical multiclass classification of software vulnerabilities. In *2021 IEEE 8th International Conference on Data Science and Advanced Analytics (DSAA)*, pp. 1–12. IEEE, 2021.

Gelei Deng, Yi Liu, Víctor Mayoral-Vilches, Peng Liu, Yuekang Li, Yuan Xu, Tianwei Zhang, Yang Liu, Martin Pinzger, and Stefan Rass. {PentestGPT}: Evaluating and harnessing large language models for automated penetration testing. In *33rd USENIX Security Symposium (USENIX Security 24)*, pp. 847–864, 2024.

Feng Dong, Liu Wang, Xu Nie, Fei Shao, Haoyu Wang, Ding Li, Xiapu Luo, and Xusheng Xiao. {DISTDET}: A {Cost-Effective} distributed cyber threat detection system. In *32nd USENIX Security Symposium (USENIX Security 23)*, pp. 6575–6592, 2023.

Min Du, Feifei Li, Guineng Zheng, and Vivek Srikumar. Deeplog: Anomaly detection and diagnosis from system logs through deep learning. In *Proceedings of the 2017 ACM SIGSAC conference on computer and communications security*, pp. 1285–1298, 2017.

Martin Eling, Rustam Ibragimov, and Dingchen Ning. Time dynamics of cyber risk. *Available at SSRN 4497621*, 2023.

FIRST. Common vulnerability scoring system version 3.1 specification document (revision 1). `https://www.first.org/cvss/v3-1/cvss-v31-specification_r1.pdf`, 2019a.

FIRST. Cvss v3.1 user guide. `https://www.first.org/cvss/v3-1/user-guide`, 2019b.

Ismayil Hasanov, Seppo Virtanen, Antti Hakkala, and Jouni Isoaho. Application of large language models in cybersecurity: A systematic literature review. *IEEE Access*, 2024.

Anders Mølmen Høst, Pierre Lison, and Leon Moonen. A systematic approach to predict the impact of cybersecurity vulnerabilities using llms. In *2025 IEEE 24th International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom)*, pp. 1598–1607. IEEE, 2025.

Kyle Ingols, Richard Chu, Richard Lippmann, Stephen Webster, and Stephen Boyer. Model-based analysis of configuration vulnerabilities. In *Proceedings of the 2009 Workshop on Cyber Security Experimentation and Test*, pp. 1–6, 2009.

Jay Jacobs, Sasha Romanosky, Benjamin Edwards, Michael Roytman, and Idris Adjerid. Exploit prediction scoring system (epss). https://arxiv.org/abs/1908.04856, 2019.

Jay Jacobs, Sasha Romanosky, Benjamin Edwards, Idris Adjerid, and Michael Roytman. Exploit prediction scoring system (epss). *Digital Threats: Research and Practice*, 2(3):1–17, 2021a.

Jay Jacobs, Sasha Romanosky, Benjamin Edwards, Michael Roytman, and Idris Adjerid. Exploit prediction scoring system (epss). *ACM Digital Threats: Research and Practice*, 2021b. doi: 10.1145/3436242.

Bo Nørregaard Jørgensen and Zheng Grace Ma. Cybersecurity and resilience of smart grids: A review of threat landscape, incidents, and emerging solutions. *Applied Sciences*, 16(2):981, 2026.

Gustav Keppler, Jeremy Kunz, Veit Hagenmeyer, and Ghada Elbez. Evaluating large language models in cybersecurity knowledge with cisco certificates. In *Nordic Conference on Secure IT Systems*, pp. 219–238. Springer, 2024.

Sego Lily Labs. Lily-cybersecurity-7b-v0.2. Model on Hugging Face, 2024.

Triet Huynh Minh Le, David Hin, Roland Croft, and M Ali Babar. Deepcva: Automated commit-level vulnerability assessment with deep multi-task learning. In *2021 36th IEEE/ACM International Conference on Automated Software Engineering (ASE)*, pp. 717–729. IEEE, 2021.

Side Liu, Guojun Peng, Haitao Zeng, and Jianming Fu. A survey on the evolution of fileless attacks and detection techniques. *Computers & Security*, 137:103653, 2024.

Mandiant. Mandiant. https://www.mandiant.com/, 2004.

Gary McGraw, Richie Bonett, Harold Figueroa, and Katie McMahon. 23 security risks in black-box large language model foundation models. *Computer*, 57(4):160–164, 2024.

Yuqiao Meng, Luoxi Tang, Feiyang Yu, Jinyuan Jia, Guanhua Yan, Ping Yang, and Zhaohan Xi. Uncovering vulnerabilities of llm-assisted cyber threat intelligence. *arXiv preprint arXiv:2509.23573*, 2025a.

Yuqiao Meng, Luoxi Tang, Feiyang Yu, Xi Li, Guanhua Yan, Ping Yang, and Zhaohan Xi. Benchmarking llm-assisted blue teaming via standardized threat hunting. *arXiv preprint arXiv:2509.23571*, 2025b.

Thu Huong Nguyen, Cédric Pruski, and Marcos Da Silveira. Deepening citation understanding in scientific literature via llm-powered context extraction. *Scientometrics*, pp. 1–32, 2026.

Offensive Security. Exploit Database (Exploit-DB). https://www.exploit-db.com/, 2025.

Xinming Ou, Sudhakar Govindavajhala, Andrew W Appel, et al. Mulval: A logic-based network security analyzer. In *USENIX security symposium*. Baltimore, MD, 2005.

Puya Pakshad, Maurice Dawson, Enkel Hoxha, and Abdul Hadi Khan. Llm-based analysis for discovering cybersecurity patterns in public administration cyber events. In *International conference KNOWLEDGE-BASED ORGANIZATION*, volume 32, pp. 1–10. Nicolae Balcescu Land Forces Academy, 2026.

Nopphon Poolsappasit, Rinku Dewri, and Indrajit Ray. Dynamic security risk management using bayesian attack graphs. *IEEE Transactions on Dependable and Secure Computing*, 9(1):61–74, 2012.

Renascence Tarafder Prapty, Ashish Kundu, and Arun Iyengar. Using retriever augmented large language models for attack graph generation. *arXiv preprint arXiv:2408.05855*, 2024.

Muthu Ramachandran. Prompt engineering for secure ai design and cyber defense. In *Guide to AI for Cybersecurity*, pp. 185–210. Springer, 2026.

José Carlos Sancho, Andrés Caro, Mar Ávila, and Alberto Bravo. New approach for threat classification and security risk estimations based on security event management. *Future Generation Computer Systems*, 113:488–505, 2020.

Davide Sanvito, Giovanni Arriciati, Giuseppe Siracusano, Roberto Bifulco, and Michele Carminati. Autocvss: Assessing the performance of llms for automated software vulnerability scoring. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing: Industry Track*, pp. 564–575, 2025.

A Sasikrishna, J Subhash, T Balachander, et al. Llm-orchestrated multi-agent framework for log-centric cyber threat detection and analysis. In *2026 7th International Conference on Intelligent Communication Technologies and Virtual Mobile Networks (ICICV)*, pp. 1490–1497. IEEE, 2026.

Mustafizur R Shahid and Hervé Debar. Cvss-bert: Explainable natural language processing to determine the severity of a computer security vulnerability from its description. In *2021 20th IEEE International Conference on Machine Learning and Applications (ICMLA)*, pp. 1600–1607. IEEE, 2021.

Oleg Sheyner, Joshua Haines, Somesh Jha, Richard Lippmann, and Jeannette M Wing. Automated generation and analysis of attack graphs. In *Proceedings 2002 IEEE Symposium on Security and Privacy*, pp. 273–284. IEEE, 2002.

Yaron Singer. Foundation-sec-8b: Cisco foundation ai's first open-source security model. `https://blogs.cisco.com/security/foundation-sec-cisco-foundation-ai-first-open-source-security-model`, 2025.

Blake Strom, Andy Applebaum, Doug Miller, Ken Nickels, Adam Pennington, and Cody Thomas. Mitre att&ck: Design and philosophy. Technical report, MITRE Corporation, 2018.

Rafael Uetz, Marco Herzog, Louis Hackländer, Simon Schwarz, and Martin Henze. You cannot escape me: Detecting evasions of {SIEM} rules in enterprise networks. In *33rd USENIX Security Symposium (USENIX Security 24)*, pp. 5179–5196, 2024.

VirusTotal. Virustotal. `https://www.virustotal.com/`, 2004.

Joshua S White, Thomas Fitzsimmons, and Jeanna N Matthews. Quantitative analysis of intrusion detection systems: Snort and suricata. In *Cyber sensing 2013*, volume 8757, pp. 10–21. SPIE, 2013.

Xian Zhan, Lingling Fan, Sen Chen, Feng We, Tianming Liu, Xiapu Luo, and Yang Liu. Atvhunter: Reliable version detection of third-party libraries for vulnerability identification in android applications. In *2021 IEEE/ACM 43rd International Conference on Software Engineering (ICSE)*, pp. 1695–1707. IEEE, 2021.

Chenxi Zhang, Xin Peng, Chaofeng Sha, Ke Zhang, Zhenqing Fu, Xiya Wu, Qingwei Lin, and Dongmei Zhang. Deeptralog: Trace-log combined microservice anomaly detection through graph-based deep learning. In *Proceedings of the 44th international conference on software engineering*, pp. 623–634, 2022.

Xin Zhou, Sicong Cao, Xiaobing Sun, and David Lo. Large language model for vulnerability detection and repair: Literature review and the road ahead. *ACM Transactions on Software Engineering and Methodology*, 34(5):1–31, 2025.

ZySec AI. Zysec-7b. Model on Hugging Face, 2024.

A AI USE STATEMENT

In this work, we used generative AI tools mainly to improve the readability of the manuscript through language polishing and grammar checking. We did not use generative AI tools to generate synthetic datasets (note that the "evidence synthesis" in our topic means "evidence integration", not fabricating data); develop theoretical models or conceptual frameworks (besides that this paper studies on LLM-powered system, rather than using generative AI as tools to develop those systems); formulate mathematical claims; provide ingredients for proving mathematical claims; assist in writing proofs; propose or refine research hypotheses; design or provide feedback on research methodology or experiments; implement the proposed methods; perform translation; clean or reformat datasets; conduct qualitative or thematic data analysis; or interpret experimental results. All AI-assisted language edits were reviewed by the authors. We take responsibility for the final content of this work, including any text produced or edited with the aid of generative AI.

B REPRODUCIBILITY STATEMENT

We provide an anonymized implementation and experimental materials at `https://anonymous.4open.science/r/LLM-Prioritization-01D2/`. The complete POLAR workflow and algorithms are described in §3, with experimental settings and evaluation protocols reported in §4. Appendix D provides implementation instructions for evidence synthesis, CVSS assessment, temporal exploitation analysis, and mitigation recommendation, while the experimental appendix documents data collection, quality control, dataset statistics, and additional evaluations. Together with the released code, these materials are intended to support reproduction of the reported results.

C LIMITATIONS

Our evaluation focuses on publicly available vulnerability and cyber threat intelligence sources and therefore does not cover organization-specific private telemetry, asset inventories, or proprietary threat feeds. We study a 30-day exploitation horizon and the mitigation categories considered in our evaluation; other operational environments may use different horizons or remediation workflows. These settings define the scope of our current evaluation, while the evidence-synthesis workflow can be adapted to additional sources and downstream security tasks.

D COMPLEMENTARY DETAILS OF POLAR

We provide implementation details for the evidence synthesis, threat prioritization, and mitigation components introduced in §3.

D.1 REAL-WORLD CYBER EVIDENCE SYNTHESIS

The evidence synthesis stage converts raw cyber reports into the threat-centric evidence $z_k = (\tau_k, \mathbf{m}_k, \mathcal{E}_k)$ used throughout POLAR. We first separate entangled threat descriptions and then enrich each threat with source-linked evidence and standardized metadata.

D.1.1 THREAT DISENTANGLEMENT

A single CTI report can mention several vulnerabilities, campaigns, or attack behaviors. We use an LLM to separate these descriptions while preserving the context associated with each threat.

Instruction: Threat Instance Disentanglement

System role. You are a cybersecurity analyst processing cyber threat intelligence reports. Separate the input into distinct threat instances.

Requirements.

- Identify explicit threat indicators such as CVE IDs, affected products, malware, infrastructure, attack techniques, and timestamps.

- Create one threat instance for each distinct vulnerability, malware activity, or campaign described in the report.
- Preserve the local context of each instance, including the affected component, exploitation behavior, required access, deployment conditions, and relevant source text.
- Do not merge two threats only because they appear in the same report.
- Do not add facts that are absent from the input.

Input.
{RAW_INCIDENT_TEXT}
Output.
Return a JSON array. Each entry contains:

- `cve_ids` if present,
- `vendor`, `product`, and `affected_component`,
- `attack_behavior`,
- `exploitation_conditions`,
- `timestamps` if present,
- `supporting_spans` copied from the input.

Each returned entry becomes a canonical threat instance τ_k. We keep the supporting spans because later stages use them as evidence for severity, exploitation, and mitigation analysis.

D.1.2 Evidence Metadata Construction and Enrichment

After disentanglement, POLAR builds the evidence set $\mathcal{E}_k$ and metadata $\mathbf{m}_k$ for each threat. The implementation follows four simple steps.

Entity linking and evidence clustering. We first link mentions that refer to the same threat. Exact CVE matches provide the strongest key when available. When a CVE is absent, we use the affected vendor/product, component, exploitation behavior, and nearby timestamps to associate reports. Repeated descriptions from different sources are grouped under the same threat while their source information is preserved.

Event decomposition and evidence roles. Long descriptions are split into evidence units that describe one relevant observation, such as a vulnerability property, PoC release, exploitation report, or mitigation statement. Each evidence unit follows the representation $e_{k,r} = (s_{k,r}, u_{k,r}, t_{k,r}, c_{k,r})$, where $s_{k,r}$ is the supporting span, $u_{k,r}$ is its source, $t_{k,r}$ is its timestamp when available, and $c_{k,r}$ records its role. We use four main roles: `identity`, `severity`, `exploitation`, and `mitigation`.

External evidence enrichment. We query external sources using the CVE ID and available aliases of τ_k. The main sources are:

- **NVD/CVE records** for canonical vulnerability descriptions, affected products and versions, disclosure dates, and references (CVE Program, 2024);
- **MITRE ATT&CK** for relevant attacker techniques and behavioral mappings (Strom et al., 2018);
- **CISA KEV** for confirmed exploitation status and associated remediation guidance (Cybersecurity and Infrastructure Security Agency (CISA), 2024);
- **Exploit-DB** and other public exploit sources for PoC availability and publication dates (Offensive Security, 2025);
- **VirusTotal** for malware, URL, file, and related threat observations (VirusTotal, 2004);
- **vendor advisories, CERT bulletins, CTI reports, and available telemetry** for affected configurations, exploitation updates, and remediation context.

The standardized metadata $\mathbf{m}_k$ contains fields that can be represented directly, such as the CVE ID, vendor/product, affected versions, affected component, disclosure date, and ATT&CK mappings. Free-text observations remain in $\mathcal{E}_k$ with their provenance.

Temporal relations and conflicting evidence. Timestamped evidence is normalized to a common date format and ordered by observation time. We retain simple relations such as the same threat appearing across sources and one observation occurring before another. Conflicting claims are not silently overwritten. Both claims remain in $\mathcal{E}_k$ with their sources and dates. For canonical fields such as affected versions or patch availability, we give preference to current vendor or government guidance and mark older information as superseded when the source explicitly indicates an update.

These steps produce the shared representation $z_k = (\tau_k, \mathbf{m}_k, \mathcal{E}_k)$. Task-specific views such as $\mathcal{E}_k^{(\mathrm{AV})}$ or $\mathcal{E}_k^{\mathrm{temp}}$ are selected from this same evidence set.

Example: Building a Print Spooler evidence metadata

Raw report i_j. A report describes both local privilege escalation and remote Print Spooler exploitation through crafted RPC requests.

Disentanglement. $\phi_{\mathrm{LLM}}(i_j)$ separates the report into distinct threat instances. For the remote exploitation path, τ_k keeps the Print Spooler component, remote access condition, crafted RPC behavior, and resulting code execution.

Metadata $\mathbf{m}_k$. POLAR links the instance to its CVE record, affected Windows versions, disclosure date, and relevant ATT&CK mappings.

Evidence $\mathcal{E}_k$. Example evidence units include "authenticated remote attacker" from a vendor advisory with role `severity`, "crafted DCERPC request" with role `severity`, and a dated KEV observation with role `exploitation`. Together, τ_k, $\mathbf{m}_k$, and $\mathcal{E}_k$ form z_k for the downstream analyses.

D.2 Threat Prioritization Details

Threat prioritization uses two views of the same evidence metadata: metric-specific evidence for CVSS assessment and temporally ordered evidence for exploitation analysis.

D.2.1 CVSS Metric Assessment

For each CVSS metric i, POLAR selects $\mathcal{E}_k^{(i)} \subseteq \mathcal{E}_k$ and provides the selected evidence, its provenance, and the official CVSS definition to the LLM. The model returns the categorical value and a short rationale pointing to the decisive evidence. Once all base metrics are assigned, we compute the final score using the official CVSS scoring procedure.

The following rules define the evidence used for each metric.

Attack Vector (AV). We look for evidence describing how the vulnerable component is reached. Network services and remote requests support `Network`; communication limited to a shared local network supports `Adjacent`; execution that requires local system access supports `Local`; and physical access supports `Physical`. Protocol and deployment information in $\mathbf{m}_k$ is used to confirm the access path.

Attack Complexity (AC). We check whether successful exploitation depends on conditions outside the attacker's control. Race conditions, rare system states, or required non-default configurations can support `High`. Normal attacker actions such as crafting a request or hosting a payload do not by themselves increase AC. If the documented exploitation path works under ordinary and reproducible conditions, we assign `Low`.

Privileges Required (PR). We determine the minimum privileges needed before exploitation begins. Explicit terms such as "unauthenticated", "authenticated user", or "administrator" are mapped to `None`, `Low`, or `High`. Privileges obtained after successful exploitation are not used to determine PR.

User Interaction (UI). We check whether successful exploitation requires an action from another user. Opening a malicious document or clicking a link supports `Required`. Exploitation triggered directly by attacker-controlled requests supports `None`.

Scope (S). We determine whether exploitation can affect a security authority different from the vulnerable component. Evidence about sandbox escape, cross-tenant effects, privilege-domain transitions,

or impact on another security boundary is used to distinguish `Changed` from `Unchanged`. When the available evidence only describes effects within the same security authority, we use `Unchanged`.

Confidentiality, Integrity, and Availability (C/I/A). We assess the direct impact of successful exploitation. Information disclosure supports confidentiality impact; unauthorized modification or code execution supports integrity impact; and crashes or service disruption support availability impact. Each dimension is assigned independently according to the extent of the documented impact.

Handling conflicting evidence. When sources disagree, POLAR checks whether the statements describe different versions, configurations, or exploitation paths. Metric assignments use the path and conditions supported by the evidence being scored and follow the official CVSS guidance when several valid paths exist. The model is instructed to cite the evidence that determines each final metric.

Example: Evidence-grounded CVSS assessment

For the Print Spooler threat above, "authenticated remote attacker" and RPC/DCERPC evidence form part of $\mathcal{E}_k^{(\mathrm{AV})}$ and support **AV=Network**. The same authentication cue contributes to $\mathcal{E}_k^{(\mathrm{PR})}$ and supports **PR=Low**. If the exploitation path requires no victim action, the corresponding evidence supports **UI=None**. The remaining metrics are determined from their own evidence views before the official CVSS procedure computes s_k.

D.2.2 TEMPORAL EXPLOITATION ASSESSMENT

The exploitation component estimates how likely a vulnerability is to be observed in exploitation within the next 30 days. We construct the temporal evidence view from evidence available at the analysis time only.

Evidence sources. We collect exploitation-related observations from the following sources.

- **Exploit-DB:** PoC availability, exploit type, and publication date (Offensive Security, 2025).
- **CISA KEV:** confirmed exploitation status and KEV addition date (Cybersecurity and Infrastructure Security Agency (CISA), 2024).
- **VirusTotal:** first-seen malware or related file/URL observations when they can be linked to the threat (VirusTotal, 2004).
- **Vendor advisories and CERT bulletins:** dated updates concerning exploitation, PoC availability, patches, and observed campaigns (Computer Emergency Response Team Coordination Center (CERT/CC), 1988).
- **CTI reports and available telemetry:** dated observations of scanning, malware activity, exploitation attempts, or campaign adoption.

Timeline construction. For each source, we extract a short event record containing its date, event type, source, and supporting text. Dates are normalized, duplicate observations are merged while preserving all sources, and events are sorted chronologically. Evidence published after the analysis cutoff is excluded. The resulting $\mathcal{E}_k^{\mathrm{temp}}$ may contain events such as disclosure, PoC publication, KEV listing, malware observation, or exploitation in the wild.

Exploitation estimation. The LLM receives z_k, the CVSS representation $\mathbf{v}_k$, and the ordered temporal evidence. It is instructed to consider the order and spacing of events, evidence of weaponization, deployment exposure, patch availability, and observed attacker interest. CVSS severity alone is not treated as evidence of active exploitation.

Instruction: Temporal Exploitation Assessment

System role. Estimate the probability that the vulnerability will be observed in exploitation within the next 30 days.

Input.

- threat evidence metadata z_k;
- CVSS metric vector $\mathbf{v}_k$;

- chronologically ordered exploitation evidence $\mathcal{E}_k^{\text{temp}}$.

Reasoning requirements.

- Use only evidence available before the analysis cutoff.
- Consider how quickly PoCs or weaponized exploits appeared after disclosure.
- Give strong weight to confirmed KEV or in-the-wild exploitation evidence.
- Consider whether the affected service is commonly exposed and whether patches are already available.
- Do not infer exploitation only from a high CVSS score.

Output.
Return:

- `probability`: a value in $[0, 1]$;
- `supporting_events`: the decisive dated observations;
- `rationale`: a short explanation of the estimate.

The returned probability is p_k. Together with z_k and s_k, it forms the prioritization metadata $r_k = (z_k, s_k, p_k)$ used for threat ordering.

D.3 Mitigation Recommendation Details

The mitigation component first retrieves source-linked defensive actions and then orders the applicable actions using the threat metadata produced during prioritization.

D.3.1 Candidate Mitigation Retrieval

For each z_k, we query the remediation corpus $\mathcal{D}_{\text{rem}}$ using the CVE ID, affected product and version, component, and relevant exploitation context. Official vendor guidance is checked first, followed by CISA KEV, NVD references, MITRE ATT&CK, maintainer documentation, CERT bulletins, and trusted security advisories.

We organize retrieved material into the four categories used in our evaluation:

- **CVE-to-patch mappings:** fixed versions, patch or hotfix identifiers, release dates, and applicability;
- **ATT&CK-aligned mitigations:** defensive techniques and controls relevant to the observed attack path;
- **mitigation notes or workarounds:** temporary configuration changes, feature disabling, segmentation, isolation, or other compensating controls;
- **vendor advisories:** official remediation guidance and supporting references.

Instruction: Mitigation Knowledge Retrieval

System role. Retrieve authoritative mitigation actions for the supplied threat metadata.

Input.
A threat metadata z_k containing the CVE ID, affected product/version, affected component, relevant exploitation evidence, and available remediation context.

Source order.

1. vendor advisories and official patch bulletins;
2. CISA KEV mitigation guidance;
3. NVD references;
4. MITRE ATT&CK mitigation knowledge;
5. official maintainer or project documentation;

6. trusted CERT or security advisories.

For each candidate action, return:

- action type;
- recommended action;
- affected and fixed versions when available;
- source and publication/update date;
- applicability conditions;
- operational notes or limitations;
- whether the guidance supersedes an older action.

Output.
Return four groups: `CVE-to-Patch`, `ATT&CK Mitigation`, `Mitigation Notes/Workarounds`, and `Vendor Advisories`.

Normalization and applicability filtering. Semantically equivalent actions from different sources are merged into one candidate while all supporting sources are retained. If a newer vendor advisory explicitly replaces an older patch or workaround, the older entry is marked as superseded. Actions that do not apply to the affected product or version in z_k are removed. Temporary workarounds remain available when a patch cannot yet be applied.

The resulting threat-specific candidate set is $\mathcal{A}_k = \{a_{k,1}, \ldots, a_{k,J_k}\}$.

D.3.2 Mitigation Ordering

The final stage receives the prioritized threat metadatas $\{r_k\}$ and their candidate actions $\{\mathcal{A}_k\}$. It uses the existing threat priority together with action-level evidence to produce the ordered mitigation plan Π_{mit}. We do not introduce a separate hand-designed risk formula in this stage.

Instruction: Risk-Aware Mitigation Ordering

System role. Produce an ordered and implementable mitigation plan from the prioritized threats and their source-grounded candidate actions.

Input. For each threat:

- $r_k = (z_k, s_k, p_k)$;
- applicable candidate actions $\mathcal{A}_k$;
- available information about patch availability, workaround effectiveness, implementation effort, and operational disruption.

Ordering rules.

1. Follow the threat priority produced by the prioritization stage.
2. Prefer an applicable vendor patch or fixed release when it can be deployed safely.
3. If immediate patching is not possible, place effective vendor-supported workarounds or compensating controls earlier in the plan.
4. For threats with strong exploitation evidence or high p_k, prioritize actions that can reduce exposure immediately.
5. Consider implementation complexity and operational disruption when actions provide similar protection.
6. Do not recommend an action if its product, version, or deployment conditions do not match z_k.
7. Keep the authoritative source attached to every recommendation.

Output.
Return an ordered list with:

- `target`;
- `recommended_action`;
- `action_type`;
- `source`;
- `priority_rationale`;
- `dependencies_or_operational_notes`.

This produces the source-linked mitigation plan Π_{mit} described in §3.3. The same threat evidence therefore supports both stages of the workflow: prioritization determines which threats are urgent, and mitigation selects the actions that fit those threats.

E Additional Details of Experiment

E.1 Data Collection and Quality Control

Raw Incident Collection. We collect raw threat incidents from multiple trusted sources, including vendor advisories (e.g., Microsoft, Cisco, Oracle, Adobe), security research reports (e.g., Mandiant, Unit 42, Recorded Future), public CTI aggregators (e.g., the CISA KEV catalog (Cybersecurity and Infrastructure Security Agency (CISA), 2024) and US-CERT bulletins (Computer Emergency Response Team Coordination Center (CERT/CC), 1988)), and exploit repositories (e.g., Exploit-DB (Offensive Security, 2025) and GitHub PoCs). Each incident is preserved in its original form, including textual advisories, PDF bulletins, blog posts, and structured feeds, so that the evaluation reflects the heterogeneous evidence encountered in practice. We exclude low-confidence or non-authoritative sources and require each incident to be corroborated by at least one vendor advisory or government-maintained repository. This process provides reliable and context-rich inputs for subsequent evidence synthesis and analysis.

I: Real-World Cyber Evidence Synthesis. Our collection and quality-control procedure follows the evidence-profile fields defined in Appendix §D.1.2. We construct reference information for the threat instances and source-linked evidence used by Polar.

1. **CVE and Vulnerability Identification.** Each candidate threat is validated against the CVE and NVD databases. Vulnerabilities used in this evaluation require an official CVE identifier and corresponding NVD entry.
2. **Tactics, Techniques, and Procedures (TTPs).** Threat instances are mapped to MITRE ATT&CK tactics and techniques using advisory text and official ATT&CK references. For example, a Print Spooler vulnerability may align with T1068 (*Exploitation for Privilege Escalation*) and T1210 (*Exploitation of Remote Services*).
3. **Affected Systems and Vendors.** Vendor, product, and version mappings are verified against NVD records and vendor advisories. For each threat, we record the affected operating systems or applications, product families, versions, and relevant architectural constraints.
4. **Exploitation and Exposure Status.** We cross-check exploitation evidence using the CISA KEV catalog, public exploit repositories such as Exploit-DB and Metasploit, and vendor or CERT bulletins. KEV listings and public exploit records are retained as separate source-linked observations because they capture different forms of exploitation evidence.
5. **Disclosure and Advisory Context.** Each threat is enriched with disclosure information, including the initial disclosure source and the chronology of advisory publication, exploit release, patch issuance, and subsequent updates when available.

II: CVSS Severity Assessment. Reference severity labels are obtained from official NVD CVSS v3.1 vectors. For each threat instance, we retrieve the categorical vector (`AV/AC/PR/UI/S/C/I/A`) and its corresponding base score. We apply the following quality-control rules:

1. If multiple CVSS entries exist because of revisions or vendor-specific scoring, we use the latest NVD-assigned vector as the reference.

2. Ambiguous or disputed CVSS assignments are flagged and excluded from evaluation.
3. We retain both categorical metric values (e.g., `AV:N`) and numerical severity scores for evaluation.

III: Temporal Exploitation Assessment. Reference exploitation likelihood is obtained from the FIRST.org Exploit Prediction Scoring System (EPSS). For each CVE, we retrieve the EPSS probability and percentile associated with a fixed evaluation date. EPSS estimates exploitation within the following 30 days and therefore matches the primary prediction horizon used by POLAR. To ensure consistent evaluation:

- EPSS scores are collected using a fixed snapshot synchronized with the evaluation window.
- CVEs without an available EPSS score are excluded from exploitation-probability evaluation.
- We retain both the raw probability and percentile rank for absolute and relative comparisons.

IV: Mitigation Recommendation. Reference mitigation items are collected from authoritative sources and organized into the same four categories used by POLAR: CVE-to-patch mappings, ATT&CK-aligned mitigations, mitigation notes or workarounds, and vendor advisories.

1. **CVE-to-Patch Mapping.** We retrieve patch records linked to each CVE and cross-check them against vendor advisory portals such as Microsoft MSRC, Cisco PSIRT, and Oracle CPU. Each entry records affected versions, fixed releases or patch identifiers, and publication dates. We retain vendor-confirmed or NVD-linked records and explicitly mark superseded patches.
2. **ATT&CK-Aligned Mitigation.** Defensive actions are collected from MITRE ATT&CK's mitigation catalog and associated with the relevant ATT&CK techniques. For example, T1210 (*Exploitation of Remote Services*) may be associated with M1031 (*Network Intrusion Prevention*) and M1051 (*Update Software*). We retain mappings supported by official ATT&CK references or corresponding vendor guidance.
3. **Mitigation Notes and Workarounds.** We collect remediation guidance from the CISA KEV catalog and other authoritative advisories. For KEV-listed vulnerabilities, we record remediation deadlines, required actions, and interim mitigations when available. Conflicting dates or instructions are resolved using the latest vendor-issued guidance.
4. **Vendor Advisories.** Vendor advisories provide the primary reference for patches, hotfixes, and configuration workarounds. We record the official advisory identifier, recommended remediation, applicable product versions, and documented operational side effects when available. Only advisories hosted on official vendor portals are used unless a secondary source is explicitly linked by the vendor.

Data Statistics. Table 3 summarizes the collected incidents and corresponding evidence used across the evaluation stages.

E.2 Additional Results and Discussions

E.2.1 Evidence Synthesis Effectiveness

Sources of performance gains. A closer inspection of POLAR outputs shows that gains in evidence synthesis mainly come from grounding raw reports in authoritative external sources. Three forms of evidence are especially useful:

1. Threat-to-record linking. Grounding advisory text against NVD records reduces ambiguity when several vulnerability identifiers appear together. For example, when an advisory discusses multiple Print Spooler CVEs, evidence linking allows each threat instance to retain the correct affected product and version.
2. MITRE ATT&CK mappings. ATT&CK grounding provides consistent behavioral context for otherwise ambiguous descriptions. For example, wording such as "gaining higher privileges through system calls" can be linked to T1068 (*Exploitation for Privilege Escalation*) when supported by the underlying evidence.

Table 3: Dataset statistics across the evaluation stages of POLAR. Numbers are rounded and reported in xK format.

Stage	Detailed Item	Count
	Raw threat incidents collected	12.8K
I. Evidence Synthesis	CVE-mapped instances	12.8K
	MITRE ATT&CK TTP mappings	12.8K
	Affected vendor/product metadata	12.8K
	Exploitation / exposure status annotated	12.8K
	Disclosure and advisory context enriched	11.6K
II. Severity Assessment	CVSS vectors (official NVD)	12.8K
	Metric-specific evidence sets	12.8K
	Fully scored CVSS severities	12.8K
III. Temporal Exploitation	EPSS probabilities (FIRST.org)	12.8K
	Exploit-DB / PoC references	6.8K
	Confirmed CISA KEV exploited CVEs	0.8K
IV. Mitigation Recommendation	CVE-to-patch release mappings (NVD + vendor)	9.9K
	ATT&CK mitigation mappings	5.2K
	CISA KEV mitigation notes	0.4K
	Vendor advisories (patches / workarounds)	8.7K

3. Disclosure and exploitation evidence. KEV records and disclosure chronology help distinguish confirmed exploitation from less certain reports. An advisory mentioning "public reports of exploitation," for example, is treated as confirmed in-the-wild exploitation only when supported by an authoritative source such as KEV or a vendor advisory.

Together, these sources improve the quality of the threat-centric evidence profile z_k before it is used for severity, exploitation, or mitigation analysis.

Nature of mistakes. Two recurring errors remain. First, some LLM backbones occasionally produce unsupported CVE identifiers when reports are long or contain several related vulnerabilities. Second, unsupported ATT&CK mappings can appear when the model extrapolates beyond the available evidence. Both errors arise when the model fills missing information using prior knowledge rather than remaining grounded in the supplied sources.

Detectability of unsupported outputs. These errors can usually be identified with lightweight validation:

- generated CVE identifiers can be checked against the CVE/NVD corpus;
- ATT&CK identifiers can be validated against the official ATT&CK taxonomy;
- claims of active exploitation can be checked against KEV and dated vendor advisories.

These checks complement evidence synthesis by preventing unsupported identifiers or exploitation claims from propagating to downstream assessment.

Takeaways. The gains in this stage come from linking LLM reasoning to source-grounded evidence. External validation is especially useful for entity linking, ATT&CK mapping, and exploitation status, where unconstrained generation can otherwise introduce plausible but unsupported information.

E.2.2 ABLATION STUDY

We provide additional discussion complementing the ablation results reported in Table 2 of §4.2.

Findings. Removing any component consistently reduces end-to-end performance, showing that the stages provide complementary information. Removing **Evidence Synthesis** causes substantial degradation because downstream stages lose the disentangled and source-grounded threat profile. Removing **Severity Assessment** mainly affects severity-sensitive ranking and patch alignment, while removing **Temporal Exploitation Assessment** reduces the ability to distinguish threats with similar severity but different exploitation dynamics.

Table 4: Zero-day threat prioritization and mitigation performance under partial-information settings. All results are the mean and standard deviation over five runs. All metrics except Kendall's τ are reported in percentage (%).

Method	NDCG@5	Kendall's τ	Mitigation Recall@5	Mitigation F1
Severity-only Heuristic	51.2 ± 1.2	0.403 ± 0.019	46.1 ± 1.4	42.8 ± 0.9
EPSS-inspired Ranking	53.8 ± 1.4	0.421 ± 0.015	48.7 ± 0.6	45.1 ± 1.1
Backbone LLM (no framework)	60.3 ± 1.5	0.497 ± 0.013	55.2 ± 0.6	51.8 ± 1.3
POLAR	71.2 ± 0.5	0.623 ± 0.012	68.4 ± 1.2	65.1 ± 0.5

Component interactions. The stages also reinforce one another. Temporal exploitation evidence is most useful when it is attached to a correctly identified threat, while CVSS assessment benefits from metric-specific evidence selected from the synthesized profile. Evidence synthesis therefore provides the shared representation on which both downstream assessments depend.

Implications for pipeline design. The modular design allows individual components to be updated without redesigning the full workflow. Improvements to threat disentanglement, evidence enrichment, CVSS interpretation, or temporal evidence collection can be incorporated while preserving the same threat-centric interface between stages.

E.2.3 ZERO-DAY ANALYSIS

We evaluate POLAR on zero-day vulnerabilities to study early-stage threat assessment when CVE identifiers, complete metadata, or official severity scores are unavailable at decision time. In this setting, POLAR relies primarily on raw incident descriptions and emerging cyber evidence.

Discussion. As shown in Table 4, POLAR substantially outperforms heuristic and backbone-only baselines under partial-information conditions. Although all methods degrade relative to fully documented vulnerabilities, POLAR retains stronger prioritization accuracy and mitigation recall by synthesizing the evidence that remains available. Qualitative inspection also shows that POLAR can recover useful mitigation actions and relative threat ordering before complete vulnerability records become available.

E.2.4 CVSS SEVERITY ASSESSMENT EFFECTIVENESS

Sources of POLAR's gains. The largest gains in CVSS assessment occur when advisory text contains ambiguous or incomplete cues. POLAR improves these decisions by selecting evidence relevant to each CVSS dimension:

1. *Attack Vector (AV).* Advisories may describe a vulnerability as "remotely exploitable" without clearly identifying the required access path. Linking the affected component to protocol and deployment evidence helps resolve cases such as RPC over SMB as `AV=Network`.
2. *Attack Complexity (AC).* Generic LLM predictions can overlook conditions such as race requirements, unusual configurations, or deployment dependencies. Metric-specific evidence helps distinguish these cases from ordinary low-complexity exploitation.
3. *Privileges Required (PR) and User Interaction (UI).* Explicit evidence such as "authenticated attacker" or "user must open a crafted document" is selected from the threat profile and used directly for the corresponding CVSS decision.

Consistency across metrics. The same evidence-grounded procedure also improves Scope and the confidentiality, integrity, and availability dimensions by aligning free-text consequences with the corresponding CVSS definitions.

Remaining gaps. Errors remain when advisories omit important technical details or when exploitation conditions differ across product versions. For example, vague descriptions such as "may allow elevation of privilege under certain conditions" may not contain enough information to cleanly separate privilege requirements from attack complexity. Version-specific behavior can also cause errors when the same CVE affects several platforms differently.

Takeaways. Overall, the results show that selecting source-linked evidence for each CVSS metric improves both categorical metric prediction and the resulting severity score. This supports the design of POLAR's evidence-grounded severity assessment.

E.2.5 TEMPORAL EXPLOITATION ASSESSMENT

Experimental Setting. We further study temporal exploitation assessment under two conditions: (1) different exploitation trajectories, including stable, increasing or decreasing, and abrupt-change patterns; and (2) different lengths of historical evidence supplied to the model. Table 5 reports forecasting performance across exploitation trajectories, while Figure 5 studies the effect of historical context length. The primary POLAR setting estimates exploitation within 30 days; we additionally report a 90-day horizon to examine sensitivity to longer prediction windows.

Table 5: Temporal exploitation forecasting by trend type using a one-year exploitation history to predict the next 30 and 90 days. Each cell reports POLAR-enhanced results together with the change relative to the corresponding backbone model. ↓ indicates reduced RMSE and ↑ indicates improved direction accuracy (DirAcc), which measures whether the exploitation trend is increasing, decreasing, or stable. All results are the mean and standard deviation over five runs. Results for cybersecurity-specialized agents are provided in Table 6.

Model		30 Days		90 Days	
		RMSE ($\times 10^{-3}$) ↓	DirAcc (%) ↑	RMSE ($\times 10^{-3}$) ↓	DirAcc (%) ↑
Monotonic trend of exploitation (increasing/decreasing)					
POLAR +	GE3.1	1.141 ± 0.029 (↓ 0.587)	89.6 ± 0.6 (↑ 12.1)	1.426 ± 0.053 (↓ 0.479)	82.1 ± 1.3 (↑ 9.0)
	G5	1.389 ± 0.049 (↓ 0.341)	85.2 ± 1.0 (↑ 7.8)	1.738 ± 0.055 (↓ 0.279)	78.1 ± 1.5 (↑ 5.7)
	L70B	1.622 ± 0.057 (↓ 0.108)	81.4 ± 1.0 (↑ 6.1)	2.029 ± 0.044 (↓ 0.089)	74.7 ± 1.4 (↑ 4.5)
	C4.1	1.098 ± 0.053 (↓ 0.629)	90.4 ± 0.6 (↑ 13.9)	1.373 ± 0.047 (↓ 0.515)	82.9 ± 1.2 (↑ 10.2)
	Q3	1.532 ± 0.035 (↓ 0.198)	82.6 ± 1.2 (↑ 6.4)	1.916 ± 0.071 (↓ 0.162)	75.9 ± 0.8 (↑ 4.7)
Stable trend of exploitation (no significant change)					
POLAR +	GE3.1	0.167 ± 0.006 (↓ 0.098)	48.7 ± 0.5 (↑ 8.9)	0.210 ± 0.010 (↓ 0.080)	42.2 ± 1.1 (↑ 6.9)
	G5	0.215 ± 0.007 (↓ 0.051)	43.8 ± 0.4 (↑ 5.9)	0.269 ± 0.007 (↓ 0.042)	37.9 ± 1.3 (↑ 4.6)
	L70B	0.247 ± 0.009 (↓ 0.019)	41.7 ± 0.9 (↑ 4.0)	0.309 ± 0.015 (↓ 0.015)	36.1 ± 0.5 (↑ 3.1)
	C4.1	0.163 ± 0.004 (↓ 0.103)	49.2 ± 0.9 (↑ 9.4)	0.204 ± 0.005 (↓ 0.084)	42.6 ± 1.1 (↑ 7.3)
	Q3	0.231 ± 0.006 (↓ 0.035)	42.7 ± 0.7 (↑ 3.4)	0.290 ± 0.010 (↓ 0.028)	36.9 ± 1.4 (↑ 2.6)
Abrupt change of exploitation					
POLAR +	GE3.1	61.473 ± 2.346 (↓ 40.764)	61.4 ± 0.7 (↑ 14.1)	76.385 ± 3.019 (↓ 33.402)	53.3 ± 0.8 (↑ 11.0)
	G5	75.212 ± 1.873 (↓ 27.121)	54.6 ± 1.5 (↑ 7.4)	93.389 ± 3.641 (↓ 22.219)	47.4 ± 1.5 (↑ 5.8)
	L70B	84.442 ± 1.835 (↓ 17.891)	48.9 ± 1.5 (↑ 1.7)	104.828 ± 2.636 (↓ 14.653)	42.4 ± 1.2 (↑ 1.3)
	C4.1	59.540 ± 2.626 (↓ 42.792)	62.5 ± 1.0 (↑ 15.2)	73.920 ± 2.926 (↓ 35.068)	54.2 ± 1.0 (↑ 11.8)
	Q3	82.821 ± 3.538 (↓ 19.512)	49.6 ± 0.7 (↑ 2.5)	102.841 ± 2.616 (↓ 15.981)	43.0 ± 1.0 (↑ 1.9)

Table 6 complements the results in Table 5 for cybersecurity-specialized agents.

Table 6: Additional temporal exploitation forecasting results by trend type, continuing Table 5. All results are the mean and standard deviation over five runs.

Model		30 Days		90 Days	
		RMSE ($\times 10^{-3}$)	DirAcc (%)	RMSE ($\times 10^{-3}$)	DirAcc (%)
Monotonic trend of exploitation (increasing/decreasing)					
POLAR +	FS	2.385 ± 0.099 (↓ 0.341)	61.8 ± 0.6 (↑ 0.1)	2.928 ± 0.115 (↓ 0.279)	56.2 ± 1.3 (↑ 0.1)
	LY	2.216 ± 0.100 (↓ 0.510)	63.0 ± 1.3 (↑ 0.2)	2.721 ± 0.113 (↓ 0.417)	57.5 ± 0.8 (↑ 0.2)
	ZY	2.127 ± 0.047 (↓ 0.599)	63.5 ± 0.7 (↑ 0.3)	2.612 ± 0.088 (↓ 0.490)	58.0 ± 0.9 (↑ 0.2)
Stable trend of exploitation					
POLAR +	FS	0.349 ± 0.010 (↓ 0.003)	34.1 ± 1.3 (↑ 0.2)	0.437 ± 0.014 (↓ 0.002)	29.6 ± 1.0 (↑ 0.1)
	LY	0.337 ± 0.016 (↓ 0.003)	35.3 ± 0.9 (↑ 0.2)	0.422 ± 0.014 (↓ 0.003)	30.6 ± 1.0 (↑ 0.2)
	ZY	0.324 ± 0.009 (↓ 0.004)	36.1 ± 0.8 (↑ 0.3)	0.407 ± 0.015 (↓ 0.003)	31.3 ± 1.2 (↑ 0.2)
Abrupt change of exploitation					
POLAR +	FS	102.847 ± 2.174 (↓ 13.485)	44.2 ± 1.5 (↑ 0.5)	127.693 ± 4.243 (↓ 11.041)	38.4 ± 0.9 (↑ 0.4)
	LY	99.264 ± 3.945 (↓ 17.069)	44.9 ± 0.6 (↑ 0.6)	123.208 ± 5.278 (↓ 13.980)	38.9 ± 1.6 (↑ 0.4)
	ZY	97.739 ± 1.979 (↓ 18.594)	45.4 ± 0.5 (↑ 0.6)	121.298 ± 5.887 (↓ 15.227)	39.4 ± 0.8 (↑ 0.5)

POLAR's robustness across exploitation trajectories. Across the trajectories in Table 5, POLAR consistently improves exploitation estimates. Baseline models can follow gradual increases or decreases but respond less reliably to sudden changes. POLAR performs better in these cases by connecting recent temporal signals with the surrounding exploitation history.

Longer temporal context is not always beneficial. Figure 5 shows that adding older exploitation history does not always improve prediction. Earlier observations can introduce evidence from outdated campaigns or product versions that is less relevant to the current threat state. This result suggests that temporal evidence should be selected according to the operational horizon rather than accumulated without limit. POLAR remains more robust than the corresponding baselines as additional context is introduced.

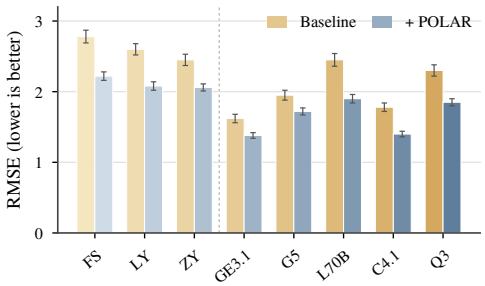

Figure 4: Overall CVSS severity scoring performance measured by RMSE.

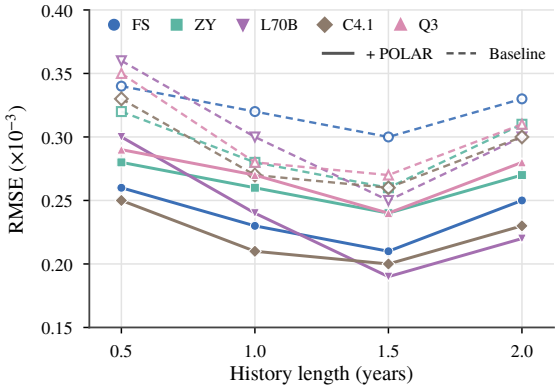

Figure 5: RMSE under varying historical context lengths for temporal exploitation assessment.

Sources of POLAR's gains. The largest improvements appear under abrupt changes in exploitation activity. In these cases, recent events such as PoC releases or new exploitation reports alter the meaning of the earlier evidence. By retaining event order and temporal context, POLAR adapts more quickly to these changes than models operating on less structured inputs.

Observed gaps. Although POLAR responds more effectively to abrupt changes, its probability estimate may still lag during the first days of a sudden exploitation surge when little precursor evidence is available.

Takeaways. These results show that temporal evidence is most useful when recent observations are interpreted in the context of the broader exploitation history. Future work may further improve responsiveness through more frequent evidence updates or adaptive selection of the temporal context window.

E.2.6 MITIGATION RECOMMENDATION EFFECTIVENESS

Experimental Setting. We evaluate the final application from two complementary perspectives: (1) the correctness of threat prioritization using NDCG@5 and Kendall's τ, and (2) mitigation retrieval accuracy measured by F1. Mitigation actions are evaluated across the four categories used throughout the paper: CVE-to-patch mappings, ATT&CK-aligned mitigations, mitigation notes, and vendor advisories.

POLAR's effectiveness in prioritization and mitigation recommendation. Results in Table 7 show consistent gains in both threat ordering and mitigation retrieval. For prioritization, higher NDCG@5 and Kendall's τ indicate that combining severity s_k with exploitation likelihood p_k produces an ordering closer to the reference urgency ranking, particularly when several threats have similar static severity.

Table 7: Evaluation of prioritization and mitigation recommendation across POLAR-enhanced cybersecurity agents and general-purpose LLM backbones. Mitigation retrieval is measured by F1 against reference items from authoritative sources. The reported values show the POLAR-enhanced result and the change (↑ or ↓) relative to the corresponding backbone model.

Model		Prioritization Correctness		Mitigation Retrieval			
		NDCG@5	Kendall's τ	CVE-to-Patch	ATT&CK Mapping	Mitigation Note	Vendor Advisory
Cybersecurity-specialized agent							
POLAR +	**FS**	0.685 (↑ 0.083)	0.553 (↑ 0.101)	0.734 (↑ 0.113)	0.635 (↑ 0.132)	0.693 (↑ 0.091)	0.625 (↑ 0.073)
	LY	0.694 (↑ 0.092)	0.564 (↑ 0.112)	0.717 (↑ 0.096)	0.624 (↑ 0.121)	0.683 (↑ 0.081)	0.616 (↑ 0.064)
	ZY	0.705 (↑ 0.103)	0.575 (↑ 0.123)	0.742 (↑ 0.121)	0.646 (↑ 0.143)	0.704 (↑ 0.102)	0.634 (↑ 0.082)
General-purpose LLM							
POLAR +	**GE**	0.793 (↑ 0.191)	0.693 (↑ 0.241)	0.844 (↑ 0.223)	0.757 (↑ 0.254)	0.845 (↑ 0.243)	0.743 (↑ 0.191)
	G5	0.873 (↑ 0.271)	0.783 (↑ 0.331)	0.943 (↑ 0.322)	0.815 (↑ 0.312)	0.903 (↑ 0.301)	0.823 (↑ 0.271)
	L70B	0.765 (↑ 0.163)	0.673 (↑ 0.221)	0.824 (↑ 0.203)	0.744 (↑ 0.241)	0.814 (↑ 0.212)	0.735 (↑ 0.183)
	C4.1	0.834 (↑ 0.232)	0.743 (↑ 0.291)	0.904 (↑ 0.283)	0.804 (↑ 0.301)	0.884 (↑ 0.282)	0.805 (↑ 0.253)
	Q3	0.806 (↑ 0.204)	0.714 (↑ 0.262)	0.872 (↑ 0.251)	0.774 (↑ 0.271)	0.863 (↑ 0.261)	0.785 (↑ 0.233)

For mitigation recommendation, POLAR improves retrieval across all four categories. The gains are especially visible for ATT&CK-aligned mitigations and mitigation notes, where recommendations depend on the exploitation path and surrounding threat context. These results show that the synthesized evidence profile supports both finding applicable defensive actions and connecting them to the threats that require attention.

Sources of gains. The improvement is most pronounced for mitigation categories that require evidence to be connected across sources. Baseline models can rely heavily on lexical similarity, while POLAR uses the threat profile to connect the affected product, exploitation behavior, and current threat evidence to corresponding defensive guidance. The same source and temporal information also helps filter outdated mitigation notes when newer vendor guidance is available.

Remaining gaps. Some errors remain for vendor-specific advisories, particularly when highly specific remediation instructions are embedded in long or inconsistently formatted documents. Sparse or conflicting severity and exploitation evidence can also affect the ordering of threats with otherwise similar profiles.

Case Study: Competing Threats. In one evaluation batch, two medium-severity vulnerabilities competed for mitigation resources: one affected a widely deployed web framework with active exploit scripts, while the other affected a peripheral library with limited exposure. A severity-driven baseline assigned similar urgency to both. POLAR ranked the first threat higher because the synthesized evidence contained stronger exploitation signals, producing an ordering consistent with the reference threat priority.

Practical Implications. The results suggest that POLAR is useful when many vulnerabilities compete for limited remediation capacity. Its shared evidence representation supports both threat ordering and the retrieval of applicable mitigation actions, reducing the need to repeat evidence collection separately for each decision.

Future Directions. Future extensions may incorporate more structured vendor feeds, improve uncertainty handling when evidence conflicts, and model deployment constraints such as patching cost or expected downtime when constructing mitigation plans.