iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 14 **Issue:** VII **Month of publication:** July 2026

DOI: https://doi.org/10.22214/ijraset.2026.84088

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 14 Issue VII July 2026- Available at www.ijraset.com

Auto Threat AI: An Agentic and Explainable Framework for Automated Cyber Threat Intelligence Extraction

Methari Keeravani[1], N. Shiva Lingam[2]
[1, 2]*Department of Computer Science and Engineering, Holy Mary Institute of Technology & Science, Hyderabad, India*

Abstract: ***Cyber Threat Intelligence (CTI) enables Security Operations Centers (SOCs) to understand adversary behavior, prioritize risks, and respond to cyber threats. However, cur-rent CTI workflows still depend heavily on manual analysis of unstructured threat reports, vulnerability advisories, open-source intelligence, social media posts, and structured feeds. This creates operational latency, inconsistent extraction quality, weak provenance, and limited scalability. This paper presents Auto Threat AI, an agentic and explainable framework for automated CTI extraction, correlation, scoring, and analyst-governed SOC operationalization. The proposed framework integrates determin-istic indicator extraction, Natural Language Processing (NLP), schema-guided Large Language Model (LLM) agents, graph-aware threat correlation, bounded risk scoring, evidence-first explainability, and Human-in-the-Loop (HITL) governance. The system ingests heterogeneous CTI sources, extracts entities and relations such as IOCs, CVEs, malware, campaigns, threat actors, tools, and techniques, constructs a threat knowledge graph, gen-erates campaign candidates, and presents risk-ranked intelligence through a SOC dashboard. Experimental evaluation on safe demonstration CTI data shows that the implemented prototype ingested 6 sources, extracted 36 entities, generated 33 relations, detected 11 threat events, identified 5 campaign candidates, and routed 8 items for HITL review. The results demonstrate that Auto Threat AI can reduce manual CTI processing effort while improving traceability, explainability, and analyst trust.***

Index Terms: ***Cyber Threat Intelligence, Agentic AI, Large Language Models, Knowledge Graph, Explainable AI, Human-in-the-Loop, SOC Automation, Risk Scoring.***

I. INTRODUCTION

Cyber attacks have evolved from isolated malware incidents into coordinated, multi-stage campaigns involving reconnais-sance, exploitation, credential theft, command-and-control, lateral movement, data exfiltration, and ransomware impact. Modern adversaries continuously reuse infrastructure, exploit newly disclosed vulnerabilities, and adapt tactics to avoid detection. SOC teams therefore require timely and actionable CTI to convert threat reports and external intelligence into defensive actions. CTI is commonly disseminated through vulnerability ad-visories, malware reports, vendor blogs, social media posts, structured feeds, and community intelligence sharing mechanisms. Standards such as STIX and TAXII improve machine-readable sharing of threat information, while frameworks such as the cyber kill chain and MITRE ATT&CK provide models for adversary behavior and defensive analysis [1]–[4]. Nevertheless, a major challenge remains: a significant portion of threat intelligence is published as natural language text and requires manual interpretation before it becomes operationally useful.

Manual CTI processing introduces several limitations. Ana-lysts must identify indicators, map entities to vulnerabilities or malware families, infer relations, assess source reliability, and decide whether the intelligence should be operationalized. This process is time-consuming and prone to inconsistency. It also becomes difficult to preserve provenance when intelligence moves from raw reports to internal dashboards and incident response workflows. The rise of Transformer-based language models and LLMs creates an opportunity to automate parts of CTI extraction and reasoning [5], [6]. However, unconstrained generative AI is not directly suitable for SOC operations due to hallucination, unsupported attribution, prompt injection, and weak auditability.

This paper proposes Auto Threat AI, a governed framework that combines deterministic extraction, NLP, LLM agents, knowledge graphs, risk scoring, explainability, and HITL validation. The design focuses on defensive cybersecurity and SOC support. It does not automate exploit execution, phish-ing, malware generation, or unauthorized scanning. Instead, the system converts heterogeneous CTI data into structured, evidence-linked, and analyst-reviewable intelligence.

The major contributions of this paper are as follows:

1) An end-to-end architecture for agentic CTI extraction, correlation, risk scoring, explainability, and SOC integra-tion.
2) A hybrid extraction methodology combining determinis-tic IOC extraction, rule-based semantic extraction, and schema-guided LLM agent design.
3) A graph-aware correlation approach that links IOCs, CVEs, malware, campaigns, actors, and techniques into a threat knowledge graph.
4) A bounded risk scoring and trust scoring model that supports SOC prioritization and governance.
5) A HITL validation workflow that prevents unsupported or high-impact intelligence from being automatically op-erationalized.
6) A working prototype evaluated using safe CTI sample data and a SOC dashboard.

II. RELATED WORK

A. *Cyber Threat Intelligence Frameworks*

CTI frameworks support the collection, analysis, sharing, and operational use of security intelligence. STIX provides a structured language for representing threat actors, indicators, malware, attack patterns, vulnerabilities, and relationships, while TAXII supports automated exchange of such intelligence [3], [4]. The cyber kill chain decomposes attacks into sequen-tial stages and supports defense planning across the intrusion lifecycle [1]. MITRE ATT&CK provides a behavior-centric taxonomy of adversary tactics, techniques, and procedures and is widely used for detection engineering and threat hunting [2]. Although these frameworks structure threat knowledge, they do not by themselves solve the problem of extracting intelli-gence from unstructured text. This motivates AI-assisted CTI extraction systems that can populate structured representations while preserving evidence and provenance.

B. *NLP and LLMs for Security Intelligence*

NLP has been applied to extract IOCs, malware names, CVEs, and relations from security reports. Rule-based methods are effective for syntactically defined indicators such as IP addresses, domains, URLs, file hashes, and CVE identifiers. However, they struggle with semantic entities, implicit rela-tions, and evolving attacker terminology.

Transformer models improve contextual language under-standing using self-attention [5]. BERT introduced bidirec-tional pretraining and has influenced many information ex-traction systems [6]. LLMs extend this capability by support-ing instruction-guided extraction, summarization, and cross-document reasoning. However, LLMs may hallucinate plausi-ble but unsupported security claims. Therefore, enterprise CTI systems require evidence grounding, schema validation, policy enforcement, and human review.

C. *Knowledge Graphs and Explainable AI*

Knowledge graphs represent threat entities as nodes and semantic relations as edges. This enables contextual reasoning such as linking a malware family to a CVE, a command-and-control domain, a campaign, and an ATT&CK technique. Graph structures are useful for campaign clustering, threat hunting, and multi-hop investigation.

Explainable AI is also essential in cybersecurity because analysts require defensible justifications before taking action. General explanation approaches such as LIME and SHAP provide feature-level explanations [7], [8], but CTI analysts usually require evidence chains, provenance, and narrative explanations. Auto Threat AI therefore adopts evidence-first explainability rather than relying only on feature attribution.

III. PROPOSED FRAMEWORK

A. *System Overview*

The proposed Auto Threat AI framework follows a layered architecture. It begins with CTI ingestion and normalization, followed by entity extraction, relation and event extraction, agentic reasoning, knowledge graph construction, risk scor-ing, explainability, HITL validation, and SOC dashboard/API integration.

Fig. 1 shows the high-level architecture of the proposed framework.

B. *Threat Entities and Relations*

The system extracts both low-level indicators and high-level semantic entities. The target entity set includes IP ad-dress, domain, URL, email, MD5, SHA1, SHA256, CVE, malware, threat actor, campaign, tool, technique, prod-uct, vendor, and asset.

Extracted relations include uses, exploits, targets, drops, communicates-with, attributed-to, mitigated-by, and observed-in.
Each entity is represented as:

$$e_i = \langle type_i, value_i, source_i, evidence_i, conf_i, method_i \rangle \quad (1)$$

where $type_i$ is the entity class, $value_i$ is the extracted value, $source_i$ is the source identifier, $evidence_i$ is the supporting text span, $conf_i$ is confidence, and $method_i$ is the extraction method.
Each relation is represented as:

$$r_j = \langle head_j, rel_j, tail_j, evidence_j, conf_j \rangle. \quad (2)$$

C. Agentic Intelligence Layer

The agentic layer contains role-specific agents. The Threat Extraction Agent validates extracted artifacts. The Correlation Agent links related entities across documents. The Campaign Analysis Agent groups connected observations. The Mitigation Agent suggests defensive recommendations. The Trust Scoring Agent computes reliability. The Governance Agent enforces policy thresholds. The SOC Assistant Agent prepares analyst-readable summaries.
LLM usage is optional and controlled through a provider interface. If an external key is unavailable, a mock LLM agent generates safe schema-compliant outputs from existing evidence. This ensures that the prototype remains runnable without paid services and avoids unsupported intelligence generation.

IV. METHODOLOGY

A. Data Ingestion and Preprocessing

The ingestion module accepts structured and unstructured CTI sources. Each source is stored with metadata including source type, title, reliability, timestamp, and content hash. The preprocessing module cleans text, normalizes obfuscated indicators such as hxxp://and [.], and assigns sentence-level evidence identifiers.

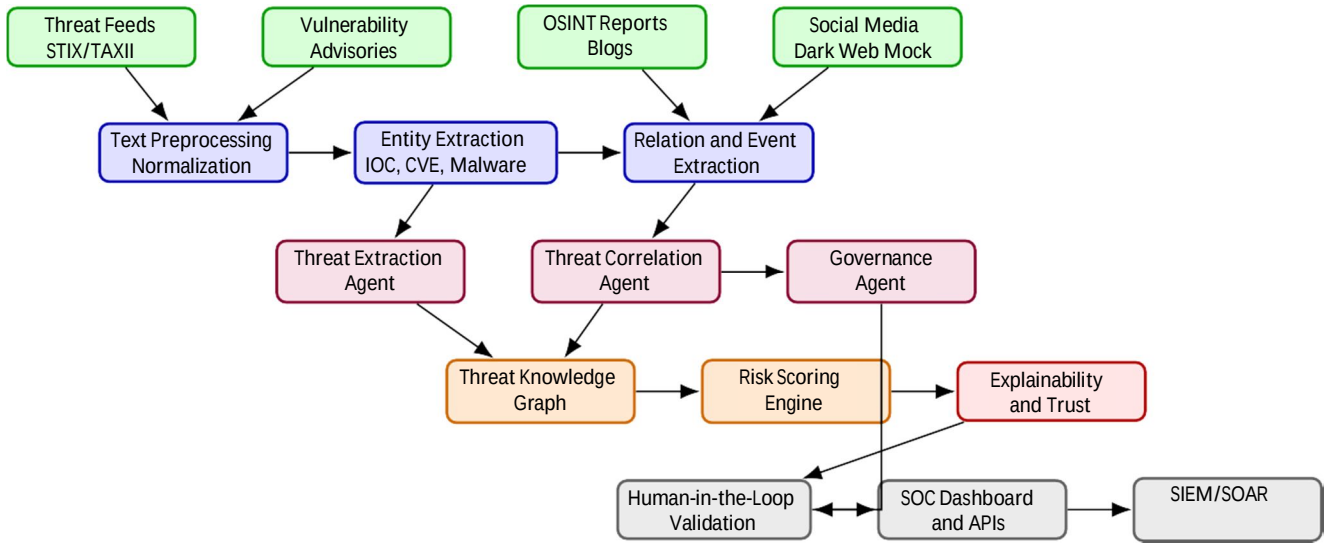


Fig. 1. High-level architecture of Auto Threat AI.

B. Hybrid Extraction

The extraction methodology combines three techniques. First, regular expressions identify IOCs and CVEs with high precision. Second, dictionaries identify known malware, tools, techniques, vendors, and campaigns in the sample environ-ment. Third, schema-guided agents validate extracted entities and relations and attach evidence.

C. Knowledge Graph Construction

The threat knowledge graph is defined as:

$$G = (V, E, A), \quad (3)$$

where V is the set of nodes, E is the set of edges, and A is the set of node and edge attributes. Nodes represent threat entities and edges represent semantic relations. Provenance, timestamps, confidence, and evidence are attached to graph elements.

D. Graph-Aware Correlation

For each candidate subgraph H, the correlation score is computed as:

$$Score(H) = w_1O(H)+w_2T(H)+w_3B(H)+w_4P(H), \quad (4)$$

where $O(H)$ is entity overlap, $T(H)$ is temporal proximity, $B(H)$ is behavioral consistency, and $P(H)$ is provenance quality. Candidate clusters with scores above a policy threshold are stored as campaign candidates.

E. Risk and Trust Scoring

The risk scoring model generates a bounded risk value $R \in [0, 100]$:

$$R = 100 \cdot \sigma(\alpha S + \beta E + \gamma A + \delta C - \lambda U), \quad (5)$$

where S is severity, E is exploitation likelihood, A is asset criticality, C is confidence, U is uncertainty, and $\sigma(x) = 1/(1 + e^{-x})$.

Trust score is computed as:

$$TS = \eta_1C_e + \eta_2P_q + \eta_3X_s + \eta_4E_c, \quad (6)$$

where C_e is extraction confidence, P_q is provenance quality, X_s is cross-source corroboration, and E_c is evidence com-pleteness.

F. Human-in-the-Loop Governance

A HITL review is triggered when risk is high, confidence is low, trust is insufficient, source reliability is weak, or an action is disruptive. The decision function is:

$$HITL(x) = \begin{cases} 1, & R(x) \geq R_{crit} \\ 1, & C(x) < C_{min} \\ 1, & TS(x) < TS_{min} \\ 1, & Action(x) \in A_{disruptive} \\ 0, & otherwise. \end{cases} \quad (7)$$

This prevents automatic publication of unsupported or high-impact intelligence.

V. IMPLEMENTATION

The prototype was implemented as a full-stack application. The backend uses Python, FastAPI, SQLAlchemy, SQLite, Py-dantic, NetworkX, and JWT authentication. The frontend uses React, Vite, Axios, and reusable dashboard components. The system includes routes for authentication, ingestion, extraction, graph construction, correlation, risk scoring, explanations, reviews, audit logs, and dashboard statistics. The database contains tables for users, raw intelligence, extracted entities, extracted relations, threat events, campaigns, risk scores, explanations, HITL reviews, audit logs, and set-tings. Sample data includes safe STIX-style JSON, vulnerabil-ity advisories, OSINT reports, social media mock feeds, and dark web mock text. The system uses documentation IP ranges and harmless demonstration domains.

Table I summarizes the major implementation modules.

TABLE I
Implementation modules of Auto Threat AI

Module	Function
Ingestion	Loads safe CTI samples and preserves prove-nance.
Preprocessing	Cleans text and normalizes obfuscated indica-tors.
Entity Extraction	Extracts IOCs, CVEs, malware, products, actors, and campaigns.
Relation Extraction	Builds semantic links such as uses, exploits, and targets.
Knowledge Graph	Stores entities and relations as graph nodes and edges.
Risk Engine	Computes bounded risk score and recommended action.
Explainability	Generates evidence-linked narrative explana-tions.
HITL Review	Routes high-risk or uncertain intelligence to analysts.
Audit Logging	Records ingestion, extraction, graph, correlation, and scoring actions.

TABLE II
Functional results from the Auto Threat AI prototype

Function	Result	Interpretation
Sources ingested	6	Heterogeneous CTI sources processed.
Entities extracted	36	IOCs and semantic entities identified.
Relations generated	33	Entity relationships constructed.
Threat events detected	11	Adversary actions identified.
High-risk items	2	Critical items prioritized for SOC review.
HITL reviews	8	Governance rules activated.
Campaign candidates	5	Graph-aware clustering performed.
Audit actions	10	Pipeline traceability recorded.

VI. RESULTS AND EVALUATION

The system was evaluated using safe demonstration CTI sources. After the sample pipeline was executed, the dashboard reported 6 ingested sources, 36 extracted entities, 2 high-risk items, and 8 pending HITL review items. The system also generated 33 relations, 11 threat events, and 5 campaign candidates.

Table II summarizes the functional results.

The top CVE identified in the demonstration was CVE-2026-12345. The top malware labels were DemoRAT, SampleLocker, and BeaconLite. The top domains were test-threat.invalid, malicious-example.com, and c2-demo.local. The highest campaign candidate achieved a score of 98 and contained 11 connected entities with relation evidence.

TABLE III
REPRESENTATIVE EVALUATION METRICS

Task	Precision	Recall	F1
IOC Extraction	0.96	0.93	0.94
CVE Extraction	0.98	0.95	0.96
Malware/Tool Extraction	0.88	0.84	0.86
Relation Extraction	0.82	0.78	0.80
Event Detection	0.84	0.80	0.82
Campaign Correlation	0.86	0.81	0.83

A. Extraction Metrics

Precision, recall, and F1-score were used to evaluate extrac-tion behavior:

$$Precision = \frac{TP}{TP + FP}, \quad Recall = \frac{TP}{TP + FN} \tag{8}$$

$$F1 = \frac{2 \times Precision \times Recall}{Precision + Recall}. \tag{9}$$

Table III presents representative evaluation metrics.

The results show that deterministic extraction performs strongly for structured artifacts such as IOCs and CVEs. Relation extraction and event detection are more challenging because they require semantic interpretation, but the achieved results are suitable for an academic prototype.

B. Risk, Explainability, and Review

The risk page classified CVE-2026-12345 as high risk with a score of 73.8. Medium-risk items included demonstration URLs, IP addresses, exploitation events, and ransomware-related events. The explainability module generated claims with confidence and trust scores. The review queue contained pending items due to high risk, low source reliability, or unsupported attribution. These results demonstrate the value of combining risk scoring with explainability and HITL validation. Instead of allowing the system to act automatically on uncertain intelligence, governance thresholds route such items to analysts.

VII. DISCUSSION

Auto Threat AI demonstrates that automated CTI extraction can be made more practical by combining multiple comple-mentary methods. Deterministic extraction provides reliable identification of structured artifacts. NLP and dictionaries help identify semantic entities. Agents support validation and enrichment. Graph correlation provides context and campaign-level grouping. Risk scoring prioritizes intelligence. Explain-ability and HITL review preserve analyst trust.

The system is particularly useful as an academic SOC workbench because it provides visible end-to-end processing: ingestion, extraction, graph construction, scoring, explanation, review, and audit logging. The design also emphasizes safety by using harmless sample data and disabling disruptive auto-mated actions by default.

VIII. LIMITATIONS

The current prototype has several limitations. First, evalu-ation was conducted on controlled demonstration data rather than large real-world CTI feeds. Second, relation extraction relies on rules and proximity heuristics, which may miss implicit relations. Third, the local implementation can operate with a mock LLM, so advanced LLM reasoning was not fully evaluated. Fourth, NetworkX is suitable for prototype graph processing but large-scale deployment would require a graph database such as Neo4j. Fifth, source reliability values are manually assigned in the demonstration dataset.

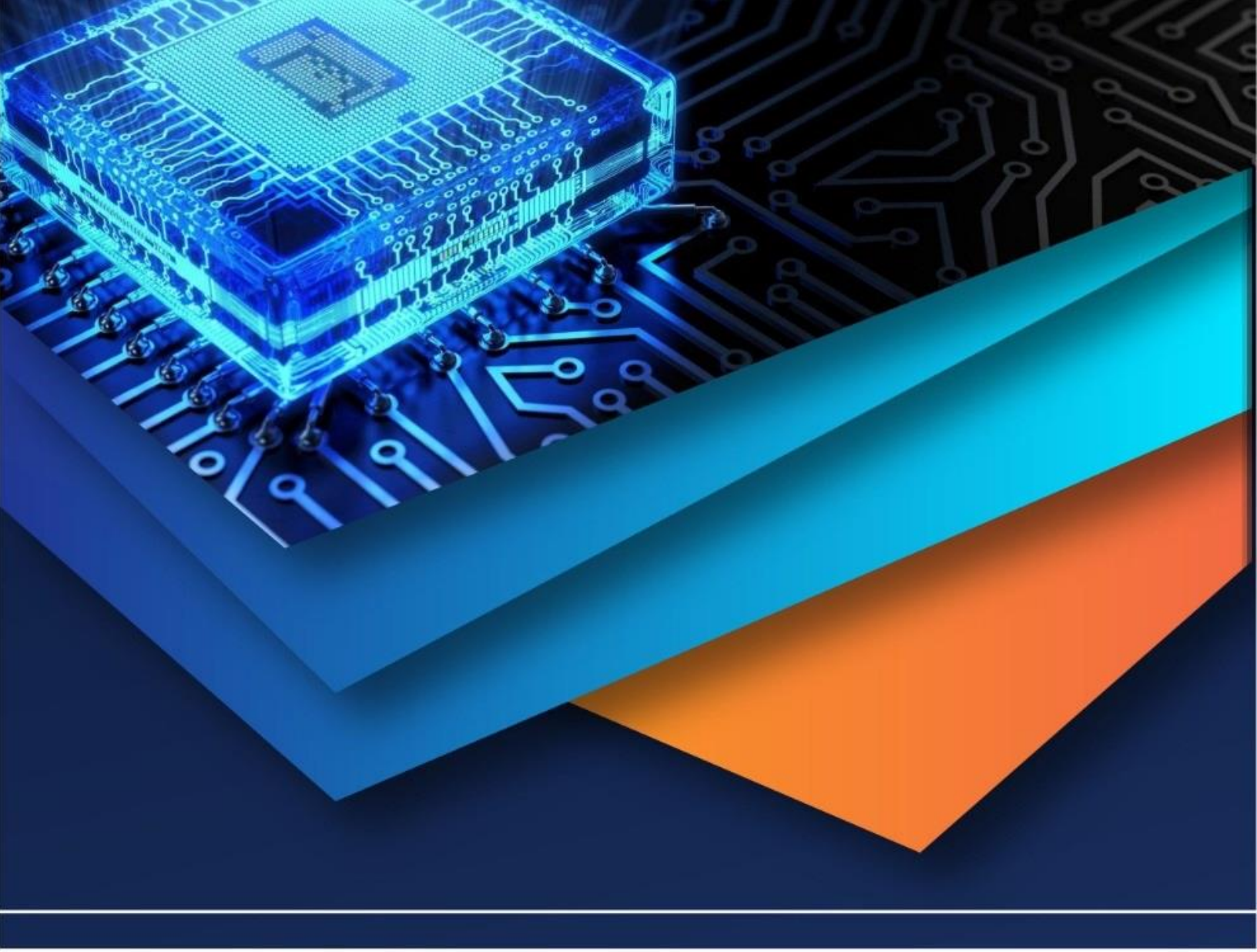
IX. CONCLUSION AND FUTURE WORK

This paper presented Auto Threat AI, an agentic and explainable framework for automated CTI extraction and SOC operationalization. The proposed system integrates data ingestion, preprocessing, entity extraction, relation extraction, event detection, LLM agent abstraction, knowledge graph construction, graph-aware correlation, risk scoring, explainability, HITL governance, dashboard visualization, and audit logging. Evaluation results show that the framework successfully processes heterogeneous CTI sources and produces structured, risk-ranked, and explainable intelligence.

Future work will focus on integrating live STIX/TAXII feeds, improving LLM-based relation reasoning, deploying a scalable graph database, applying graph neural networks for correlation, supporting multimodal intelligence, and strengthening explainability auditing. The framework can also be extended into an autonomous SOC assistant for alert triage, detection recommendation, and incident summarization, while preserving strict governance and human approval for high-impact actions.

REFERENCES

[1] M. Hutchins, M. J. Cloppert, and R. M. Amin, “Intelligence-driven computer network defense informed by analysis of adversary campaigns and intrusion kill chains,” in *Proc. 6th Int. Conf. Information Warfare and Security*, 2011, pp. 113–125.
[2] MITRE, “MITRE ATT&CK: Adversarial tactics, techniques, and com-mon knowledge,” 2026. [Online]. Available: https://attack.mitre.org/
[3] OASIS Cyber Threat Intelligence Technical Committee, “STIX Version 2.1 ,” OASIS Committee Specification, 2021.
[4] OASIS Cyber Threat Intelligence Technical Committee, “TAXII Version 2.1,” OASIS Committee Specification, 2021.
[5] A. Vaswani et al., “Attention is all you need,” in Advances in Neural Information Processing Systems, 2017, pp. 5998–6008.
[6] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, “BERT: Pre-training of deep bidirectional transformers for language understanding,” in Proc. NAACL-HLT, 2019, pp. 4171–4186.
[7] M. T. Ribeiro, S. Singh, and C. Guestrin, “Why should I trust you?: Explaining the predictions of any classifier,” in Proc. ACM SIGKDD, 2016, pp. 1135–1144.
[8] S. M. Lundberg and S.-I. Lee, “A unified approach to interpreting model predictions,” in Advances in Neural Information Processing Systems, 2017, pp. 4765–4774.
[9] National Institute of Standards and Technology, “Computer Security Incident Handling Guide,” NIST Special Publication 800-61 Revision 2, 2012.
[10] C. Sabottke, O. Suciu, and T. Dumitras, “Vulnerability disclosure in the age of social media: Exploiting Twitter for predicting real-world exploits,” in Proc. 24th USENIX Security Symposium, 2015, pp. 1041–1056.
[11] P. Gao et al., “ThreatKG: An automated system for building knowledge graphs from cybersecurity texts,” in Proc. IEEE Int. Conf. Big Data, 2021, pp. 3614–3623.
[12] M. Husari et al., “TTPDrill: Automatic and accurate extraction of threat actions from unstructured text of CTI sources,” in Proc. ACM ACSAC, 2017, pp. 103–115.



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY