An Educational and Validation Tool for Cyber Threat Intelligence Leveraging Large Language Models

Stiven Janku, Hannan Xiao* and Timmy Caris

Department of Informatics, King's College London, Strand Campus, Bush House, 30 Aldwych, London, WC2B 4BG

Abstract

Cyber Threat Intelligence (CTI) has always played a pivotal role in proactive cybersecurity. However, with the emergence of Large Language Models (LLMs), generating and disseminating false or misleading CTI has never been easier. Existing research has found that fabricated CTIs could successfully evade cybersecurity professionals, but there is a notable gap in detecting fabricated CTIs. This paper addresses how LLM-based approaches can serve as a powerful tool for validating the authenticity of reported threats. We propose a framework for evaluating text-based intelligence through a structured ranking of sources, automated keyword extraction, and a final AI-based analysis that yields a probability score to identify potential misinformation. Our evaluation using 150 CTI reports (authentic, LLM-generated, and hybrid) demonstrates strong classification performance with an overall F1-score of 0.88, achieving particularly high accuracy for completely fabricated reports while identifying partially manipulated content with moderate success. Beyond technical validation, VeraCTI serves as an educational platform for cybersecurity practitioners through its transparent, step-by-step analysis process, which can be deployed in Security Operations Centres (SOCs) to simultaneously enhance threat verification capabilities and develop analysts' critical assessment skills. By operating on the principle that "all information is false until proven", VeraCTI addresses a critical gap in current CTI validation approaches and demonstrates how AI systems can be leveraged responsibly to counter AI-generated misinformation.

Keywords

GenAI, LLM, CTI, AI, cybersecurity, cyber threat intelligence, validation, generation, cybersecurity education

1. Introduction

Cyber Threat Intelligence (CTI) plays an increasingly pivotal role in modern cybersecurity, providing the necessary foresight for organisations to anticipate, identify, and mitigate sophisticated cyber threats [1]. An effective CTI is crucial for guiding proactive defence measures, including timely incident response, informed patch management, and heightened vulnerability awareness. However, the landscape is complicated by the dual-edged nature of emerging Artificial Intelligence (AI) technologies, particularly Large Language Models (LLMs) [1]. Although LLMs offer unprecedented capabilities for processing and analysing large volumes of unstructured data inherent in CTI [2, 3], they also introduce significant risks.

The core problem this paper addresses is the capability of LLMs to generate convincing, yet potentially fabricated or misleading CTI [4, 5]. Malicious actors can leverage the same generative power that aids defenders to create and disseminate fabricated CTI at scale, designed to poison datasets, mislead security analysts, and obfuscate genuine threats [6]. This potential for AI-driven misinformation poses a substantial challenge, as ingesting unreliable CTI can lead to misallocated resources, flawed security postures, and ultimately successful breaches. The inherent uncertainty sometimes present in CTI descriptions further complicates matters, making it difficult even for advanced models to interpret intent accurately [7]. In addition, the generation and spread of fabricated CTI undermine the trust necessary for effective intelligence sharing and collaborative defence.

This challenge necessitates robust validation mechanisms capable of discerning authentic intelligence from LLM-generated fabrications. While traditional methods of CTI validation often rely on structured

Joint Proceedings of IS-EUD 2025: 10th International Symposium on End-User Development, 16–18 June 2025, Munich, Germany.

*Corresponding author.

stiven.janku@kcl.ac.uk (S. Janku); hannan.xiao@kcl.ac.uk (H. Xiao); timmy.caris@kcl.ac.uk (T. Caris)

0009-0008-9009-566X (S. Janku); 0000-0003-2273-6679 (H. Xiao); 0009-0004-9215-437X (T. Caris)

Indicators of Compromise (IoCs) and cross-referencing with established repositories, they struggle with the nuance and volume of unstructured, text-based intelligence, especially when its authenticity is questionable from the start [8]. A significant gap exists in the literature regarding robust, scalable methods for validating the *content* of text-based CTI, particularly in the face of potential LLM-driven disinformation campaigns. Existing approaches primarily focus on source reputation or structured data consistency, leaving unstructured textual claims largely unverified [5, 9, 10]. Our work aims to address this gap by proposing a framework that leverages LLMs themselves, guided by structured analysis and external corroboration, to assess the probability of falsehood in text-based CTI reports.

In this paper, we propose that LLMs themselves, when guided appropriately, can be instrumental in *validating* CTI. By leveraging their advanced natural language understanding and reasoning capabilities, LLMs can assist practitioners in assessing the credibility of threat reports, identifying inconsistencies, and flagging potential misinformation. This requires a structured approach. Simply asking an LLM whether a report is "true" is insufficient. Our work introduces VeraCTI (the name stems from a combination of two words, veracity and CTI), a framework, an educational tool, and a methodology designed to systematically evaluate text-based CTI. It employs a methodology incorporating source reliability ranking, targeted keyword extraction for semantic analysis, and an LLM-based reasoning component to produce a probabilistic score indicating the likelihood of a report being fallacious or trustworthy. This approach assumes "all information is false until proven," promoting a critical mindset essential in the current threat landscape.

Addressing the challenge of fabricated CTI also requires an educational dimension. Cybersecurity practitioners, from novices to experienced analysts, must be trained to critically evaluate AI-driven intelligence, understand the limitations of LLMs, and effectively utilise validation tools [11]. VeraCTI is thus designed partly as an educational resource, helping users understand the factors contributing to CTI credibility and fostering best practices for cross-checking AI-generated insights against verified sources. By providing a quantifiable measure of trust and highlighting potential red flags, our framework empowers security teams to make more informed decisions, integrate LLM capabilities responsibly, and ultimately strengthen their defensive position against both conventional and AI-amplified cyber threats.

The remainder of this paper is organised as follows. Section 2 reviews the existing work on LLMs in CTI and on validation approaches. Section 3 details the proposed VeraCTI framework for LLM-driven verification of CTI. Section 4 describes the implementation details. Section 5 presents preliminary results of the testing and evaluation. In Section 6, we discuss the way this tool can be integrated in education and organisations. Finally, Section 7 concludes the paper and outlines avenues for future research.

2. Related Work

The integration of AI, particularly LLMs, into cybersecurity represents a rapidly evolving research frontier [1, 11]. LLMs, with their advanced natural language understanding and generation capabilities, are increasingly being explored for various defensive and offensive cybersecurity applications [4, 8]. This section reviews the relevant literature, focusing specifically on the application of LLMs to CTI tasks and highlighting the emerging challenge of validating AI-generated or potentially falsified intelligence.

2.1. LLMs in Cyber Threat Intelligence Processing

CTI is fundamental to proactive cybersecurity, involving the collection, analysis, and dissemination of information about cyber threats [12]. Traditionally, processing CTI, often found in unstructured reports, blogs, and security advisories, has been a labour-intensive task for human analysts [13]. Recent advances demonstrate the significant potential of LLMs to automate and enhance various stages of the CTI lifecycle.

Several studies have focused on leveraging LLMs for extracting structured information from unstructured CTI sources. For instance, LLMs have been employed to identify and extract Tactics, Techniques,

and Procedures (TTPs) and other cyber-related entities [3, 14, 2]. Systems like LLM-Tikg [15] and the framework proposed by Zhang et al. [16] utilise LLMs to automatically construct Cybersecurity Knowledge Graphs (CKGs) from CTI reports, facilitating better organisation and querying of threat data [10, 13]. Other works, like aCTIon [14] and LLMCloudHunter [2], demonstrate the use of LLMs (often GPT-3.5 or GPT-4[**?**] variants) with zero-shot or few-shot prompting and specific pipelines to distil and structure information from diverse Open Source Intelligence (OSINT) sources.

Domain-specific models like SecureBERT [17] and CySecBERT [18] have also been developed to better handle the specific vocabulary and context of cybersecurity texts. Approaches like LOCALINTEL combine retrieval mechanisms (similar to Retrieval-Augmented Generation (RAG)) with LLMs to generate contextualised, organisation-specific CTI [19]. The CYLENS system further exemplifies this trend, acting as an LLM-powered CTI copilot integrating knowledge from numerous threat reports [20]. These studies generally focus on generating or structuring CTI rather than verifying its authenticity.

2.2. The Challenge of Misinformation and Fake CTI

While the ability of LLMs to process and generate CTI offers significant advantages, it concurrently introduces a critical vulnerability: the potential for generating and disseminating fake or misleading intelligence [21]. Malicious actors can exploit the same generative capabilities to craft convincing but false threat reports, aiming to poison datasets used for training AI-based defence systems or to mislead human analysts and automated tools [6]. This constitutes a sophisticated form of data poisoning attack tailored to the CTI ecosystem [22].

The generation of fake CTI using fine-tuned transformer models like GPT-2 has already been demonstrated, with studies showing that such fabricated intel can be highly convincing, even to experienced cybersecurity professionals [6]. The work by Yu and Li specifically focuses on methods for generating fake CTI using models such as GPT-Neo [23]. The inherent plausibility of LLM-generated text makes manual verification difficult and time-consuming, deepening the existing challenge of information overload in CTI [12]. Furthermore, the propagation of fake CTI through OSINT channels can undermine the credibility of legitimate sources and disrupt collaborative defence efforts.

2.3. Validation and Credibility Assessment in CTI

Given the risks associated with fake CTI, validating the authenticity and credibility of intelligence is paramount. Traditional CTI quality assessment has often focused on the reputation or trustworthiness of the *source* [24], or on the consistency of structured data [9]. However, assessing the veracity of the *content* itself, particularly unstructured text-based claims, remains an open challenge, especially with the potential for sophisticated AI-generated fakes.

Existing work on CTI credibility often relies on metadata or simple examining. For example, Tundis et al. [24] proposed a feature-driven method to assess OSINT source relevance on Twitter but did not focus on content veracity. Yang et al. [9] developed a CTI quality assessment method considering feed trustworthiness and content availability metrics (like timeliness, completeness) but did not specifically address deliberate falsification or AI-generated misinformation. While frameworks like KGV [10] integrate LLMs and Knowledge Graphs (KGs) for *credibility assessment*, their primary mechanism involves fact-checking extracted claims against a KG built from paragraphs, which may not be sufficient against LLM-generated narratives lacking direct counter-evidence in the graph. While the landscape of LLMs in cybersecurity is reviewed [11], specific methods for validating CTI content veracity are needed. The AI4CYBER framework, mentioned in [5], touches upon trustworthiness services but primarily in the context of ensuring the reliability of its *own* AI-components, rather than validating external CTI content.

3. Design

3.1. Overall System Architecture

VeraCTI is structured as a modular, event-driven pipeline that processes diverse inputs (text, files, URLs), enriches them with external threat intelligence, and generates a quantitative assessment of the likelihood that the provided CTI is misleading or potentially inaccurate (rather than purely false). Figure 1 provides an overview of the entire pipeline, beginning with front-end collection and culminating in a comprehensive threat intelligence report.

The architecture is divided into three main layers:

- First, the *data ingestion* layer receives raw CTI from various sources and ensures input quality.
- Second, the *analysis and correlation* layer employs natural language processing (NLP) for key-phrase extraction, identifying potential IoCs and Common Vulnerabilities and Exposures (CVEs), and correlating the claims with external threat feeds.
- Third, the *aggregation and reporting* layer computes a final probability score, indicating the rationale behind that score, and generates a user-friendly, structured report.

Internally, VeraCTI uses a combination of synchronous and asynchronous pipelines. Certain tasks, such as normalising textual data, must be performed in a strictly ordered manner. Other operations, such as IoC enrichment from external APIs, can run in parallel, improving efficiency when dealing with multiple indicators. Each module is designed to be stateless: after receiving normalised inputs and performing its assigned task, it outputs well-defined data structures to the subsequent module.

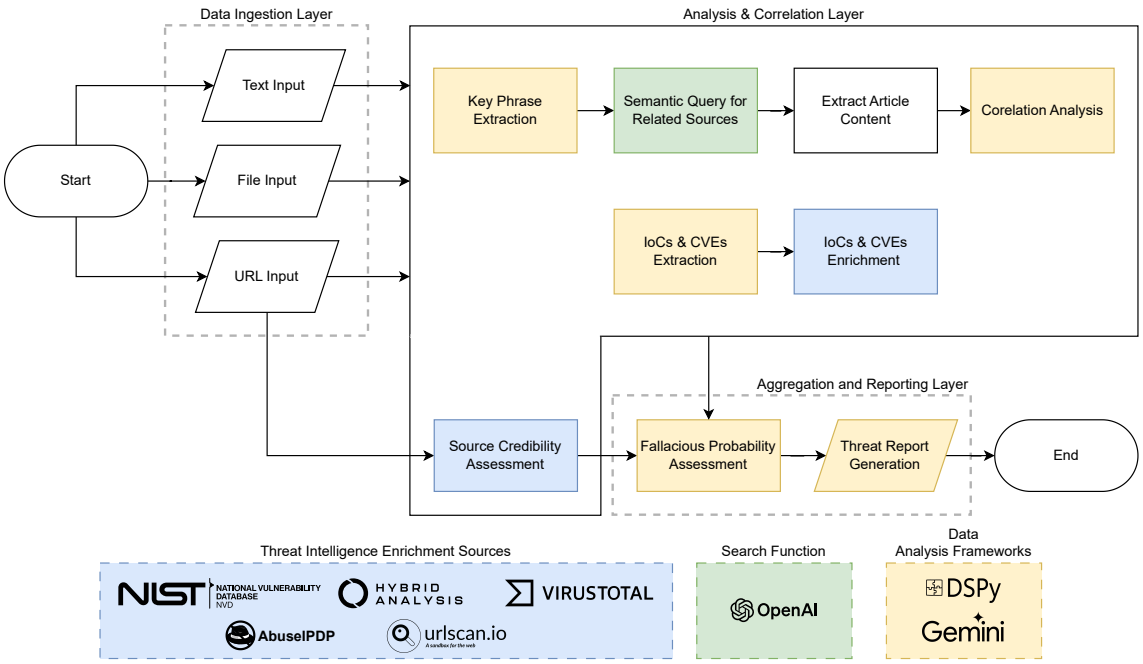

Figure 1: High-level system architecture and data flow for the VeraCTI CTI Analysis System.

3.2. Data Ingestion Layer

VeraCTI enforces a rigorous input validation process to ensure that any data entering the system is both properly formatted and has passed an initial credibility check. For textual submissions, VeraCTI imposes token limits that keep LLM-driven analysis computationally feasible. Files, most commonly PDFs, are parsed using extraction libraries (pdfminer.six)[?] to convert them into plain text. If the parser detects incomplete or corrupted content, the system flags the submission for additional scrutiny.

URLs undergo a separate validation process wherein VeraCTI queries domain-reputation services such as *urlscan.io*[**?**]. If the domain is assessed as malicious or high-risk, the system includes this low-trust signal in subsequent correlation tasks. VeraCTI then fetches the HTML content of the URL, stripping it to text via BeautifulSoup[**?**], and incorporating relevant metadata such as HTTP response codes. Through each of these methods, the data ingestion layer produces a coherent block of text accompanied by metadata (including source reputation, domain trust scores, and file integrity checks) for downstream analysis.

3.3. Analysis and Correlation Layer

Key Phrase Extraction Once data is validated and the text is normalised, VeraCTI applies NLP techniques to identify key concepts within the report. This involves an LLM or a transformer-based pipeline that scans for domain names, threat actor handles, and references to known vulnerabilities or malicious campaigns. By extracting not only direct IoCs but also contextual entities (e.g. mention of a campaign name or a technique like "credential stuffing"), the system builds a richer picture of the underlying intelligence.

Semantic Query for Related Sources VeraCTI performs a semantic query of external articles or threat feeds that might corroborate or contradict the extracted key phrases. This step helps the system compare the CTI content against existing records, increasing confidence in valid claims and reducing the risk of overlooked inconsistencies.

IoC & CVE Extraction and Enrichment After key phrase extraction, the system specifically focuses on identifying Indicators of Compromise (IoCs) such as IP addresses, URLs, file hashes, and CVE identifiers. VeraCTI re-fangs obfuscated addresses (e.g. `hxxp://` to `http://`), ensuring the extracted IoCs are in a standard format. It then queries external threat intelligence feeds, including *VirusTotal*[**?**] or *AbuseIPDB*[**?**], to gather reputation data and historical activity. If the text claims an IP address is part of a widespread phishing campaign and *VirusTotal* corroborates it with multiple malicious detections, VeraCTI increases its confidence in that portion of the intelligence.

Conversely, if external databases record an IoC as benign or do not recognise it at all, the system raises a flag indicating a possible inconsistency. Multiple signals from distinct sources reinforce or dispute the authenticity of the reported IoC. In this way, VeraCTI combines textual coherence with real-world data about previously observed malicious or benign activity, lending a structured perspective to each piece of intelligence provided.

3.4. Aggregation and Reporting Layer

VeraCTI concludes its analysis by computing a probability score that represents the likelihood a CTI report is misleading, partially false, or entirely fabricated. Each factor—including source credibility, IoC reputation, external corroboration, and text consistency—contributes to this final score through a weighted scheme. The weighting can be adapted to suit specific organisational needs, such as prioritising the trust signals from certain authoritative repositories.

The system outputs these results in a structured JSON object, providing both a numeric estimate (e.g. 0–100% fallaciousness probability) and an explanatory breakdown. This transparency benefits both automated workflows, which can quickly parse numeric thresholds, and human analysts, who can review the salient points leading to VeraCTI's classification. By separating the final score from the evidence, the system also facilitates educational use, illustrating exactly which elements are suspicious or well-supported.

4. Implementation

An automated pipeline is implemented within the VeraCTI application to analyse CTI inputs, assess their potential fallaciousness, and generate a comprehensive analysis report. The system processes

CTI provided via direct text entry, file upload, or URL submission, employing a multi-stage approach involving AI-driven analysis, external data enrichment, and structured reporting.

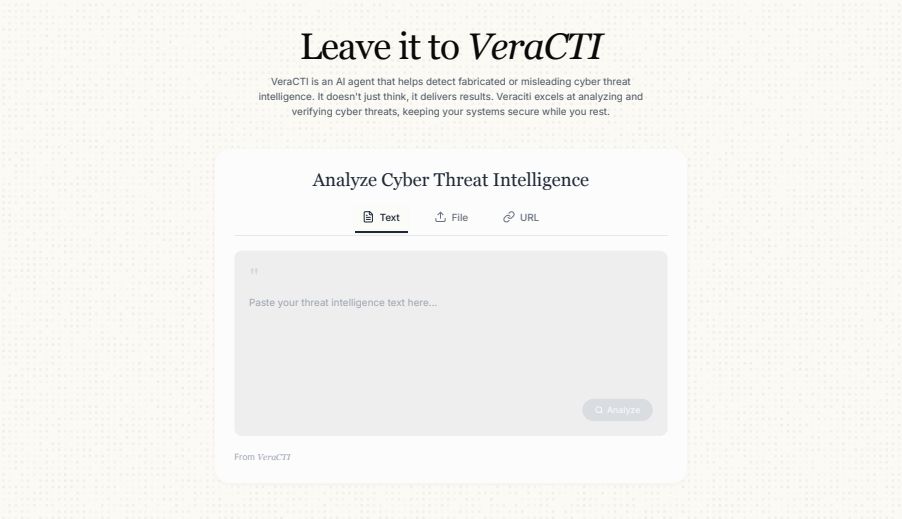

Figure 2: VeraCTI's landing page, where users can submit text, files, or URLs for CTI analysis.

4.1. Input Ingestion and Pre-processing

The pipeline initiates by receiving CTI through one of the designated frontend channels. Pre-processing steps normalise the input for consistent analysis:

Text Input: User-submitted raw text (up to a practical limit, e.g. ~65,000 tokens, with a maximum of 1 million tokens) is directly relayed to the primary text analysis engine. No significant pre-processing occurs at this stage beyond basic validation in the API route.

File Input: The file is validated and temporarily saved, using a UUIDv4-based filename. The script then uses the pdfminer.six library's `extract_text()` function to retrieve the full text content. Error handling is present for file I/O and extraction issues. The temporary file is deleted in a `finally` block within the API route after processing completes or fails.

URL Input: Basic validation (for instance, requiring the scheme be `http://` or `https://`) occurs first. The script then orchestrates source credibility check and content extraction. In **source credibility check**, the connector module's `url_scan_details()` function queries the *urlscan.io* API for domain reputation, scan results, and potential malicious verdicts. This provides an initial trust assessment of the source URL itself. In **content extraction** the `connector`'s `extract_text_from_url()` function fetches the URL's HTML content using requests[**?**] and parses the main textual body via BeautifulSoup, stripping HTML tags to provide clean text for analysis. Timeout and connection error handling are included.

4.2. AI-Driven Content Analysis and Corroboration

With the CTI text extracted, a sequence of approximately 4–5 distinct Declarative Self-improving Python DSPy [**?**] agents perform deeper analysis. These agents leverage an LLM configured within the programme (e.g., *Gemini*[**?**], `temperature=1.0`, `max_tokens=65536`, `cache=true`), utilising `dspy.ChainOfThought()` for reasoning.

Key Phrase Identification: The `CategoriesKeywords()` agent identifies and extracts up to 10 core thematic sentences or key phrases from the input CTI text. It uses the KeywordCategories Pydantic model for structured output ({`"extracted_categories"`: [`"sentence1"`, ...]}).

External Corroboration: The extracted key phrases serve as queries for the connector's `search_web_for_related()` function. This function (leveraging the configured dedicated search API) seeks external articles or reports online that could corroborate or contradict the input CTI. It returns a list of relevant URLs.

Relevance Evaluation: For each relevant external URL found, its content is extracted (similar to URL input processing). The `RelevanceCheck()` agent then compares the original CTI text against the external article's content. It outputs a structured dictionary (Result Pydantic model) detailing matches in IoCs/CVEs, contextual text similarity, reasoning for the match, and an overall relevance score (e.g., on a 1–10 scale).

4.3. IoC and CVE: Extraction and Enrichment

Extraction: The `IOCExtraction()` agent parses the input CTI text, specifically targeting IoC patterns. It aims to extract only indicators deemed relevant within the threat intelligence context, avoiding generic examples. Output uses the Extract_IoCs Pydantic model ({`"ip_addresses"`: [...], `"urls"`: [...], ...}). Defanged indicators (e.g., `hxxp://` or `1.1.1[.]1`) are re-fanged by the agent based on instructions.

Enrichment: The extracted IoCs are systematically queried against multiple external Threat Intelligence platforms using the Connector and API keys configured. Approximately five external services are potentially queried per relevant IoC type:

- **IPs:** *VirusTotal* (Reputation, Votes, WHOIS), *AbuseIPDB* (Confidence Score, History).
- **Domains:** *VirusTotal* (Reputation, Votes, WHOIS).
- **URLs:** *VirusTotal* (Submission for Scan, Analysis Stats).
- **Hashes (SHA256, MD5, SHA1):** *Hybrid Analysis*[**?**] (Verdict, Threat Score, Associated Reports).
- **CVEs:** *National Vulnerability Database*[**?**] (Common Vulnerability Scoring System (CVSS) Score, Description, References).

Each query returns structured data about the indicator's reputation, known associations, and analysis results from the respective platform. Error handling for API requests (timeouts, connection errors, HTTP errors) is implemented.

4.4. Synthesised Reporting and Fallaciousness Assessment

The culmination of the pipeline involves aggregating all findings and generating a final assessment:

Data Aggregation: All intermediate results—initial source credibility checks (for URLs), web corroboration findings (relevance scores, summaries), and detailed IoC enrichment data—are collected.

Structured Report Generation: The `Threat Intel Report()` agent receives this aggregated data and synthesises the diverse inputs into a single, structured JSON output conforming to the pre-defined scheme. It populates fields for metadata, executive summary, source analysis, content corroboration, detailed enrichment analysis per IoC type, and crucially, the fallacious probability assessment.

Fallacious Probability Assessment: A dedicated section in the final report represents the system's core judgement on the likelihood of the input CTI being misleading or fake. Conceptually, it follows:

$$\text{Fallacious Probability} \approx f\ \Big(\alpha \cdot \text{SourceCredibility} + \beta \cdot \text{ContentCorroboration} + \gamma \cdot \text{IoC Validity} + \delta \cdot \text{InternalConsistency}\Big) \tag{1}$$

where the function f maps these weighted factors to a final probability score (*e.g.*, "Low," "Medium," "High," or 0–100) and explanatory reasoning, based on the following variables.

- *SourceCredibility*: Derived from *urlscan.io* results and domain reputation (0.0–1.0).
- *ContentCorroboration*: Based on the number and relevance scores (e.g., avg. score 1–10) of external matching articles found.
- *IoC Validity*: Reflects the proportion and severity of IoCs flagged as malicious by enrichment services (*e.g.*, *VirusTotal* malicious votes > 5, *AbuseIPDB* score > 75).
- *InternalConsistency*: Assessed by the LLM for logical coherence within the CTI text itself.
- $\alpha, \beta, \gamma, \delta$: Weighting factors implicitly determined by the LLM based on the specified instructions.

5. Testing and Evaluation

5.1. Dataset

To evaluate VeraCTI's effectiveness, we constructed a balanced dataset comprising three distinct categories of CTI reports:

Authentic CTI (n=50): We collected genuine threat intelligence reports from authoritative sources including official Computer Emergency Response Team (CERT) advisories, vendor security bulletins, and established threat intelligence platforms. These reports were manually verified to ensure accuracy and relevance.

LLM-Generated Synthetic CTI (n=50): Using both GPT-4 and Gemini-Pro models, we generated fabricated threat intelligence reports. These were crafted with varying levels of sophistication, from simple fabrications to complex reports incorporating legitimate IoCs in misleading contexts. We used prompts designed to create plausible but false narratives about non-existent threats, vulnerability exploitation, or threat actor campaigns.

Hybrid CTI (n=50): We created partially modified authentic reports by manually altering key details while maintaining overall structure and context. Modifications included replacing legitimate IoCs with benign ones, exaggerating severity levels, or introducing inconsistencies in technical details while preserving the narrative flow of genuine reports.

Each dataset entry was anonymised and assigned a unique identifier to eliminate bias during evaluation. The distribution of content types (text-only, file, URL) was maintained consistently across all three categories to ensure evaluation fairness.

5.2. Evaluation

In this paper, we focus on the evaluation of classification accuracy by precision, recall and F1-Score. Precision measures how many of the items labelled as a certain class are truly in that class. Formally, it is the ratio of correctly predicted positive observations to the total predicted positive observations:

$$\text{Precision} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Positives}} \tag{2}$$

In the context of this evaluation, **True Positives** can be defined as CTI reports correctly classified as either factual or fallacious, while **False Positives** are reports that have been incorrectly classified (e.g., a fabricated report misclassified as factual). The precision value thus represents the reliability of the classifications made by VeraCTI.

Recall is defined as the proportion of items that truly belong to a class and are correctly identified:

$$\text{Recall} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Negatives}} \tag{3}$$

False Negatives are represented by reports that have not been correctly identified by the system (e.g., a fallacious report not flagged as such). The recall value therefore indicates the comprehensiveness of the system in identifying all instances of a particular class.

The F1-score is calculated as the harmonic mean of precision and recall:

$$\text{F1-Score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{4}$$

It balances precision and recall into a single number, especially useful when both metrics need to be weighed equally. If precision and recall differ significantly, the F1-score provides a combined measure that diminishes the effect of extremely high precision or recall alone. It's especially insightful in cases where a single metric is needed to convey the overall effectiveness of the model.

Table 1
Classification performance metrics for VeraCTI across different CTI categories.

CTI Category	Precision	Recall	F1-Score
Authentic CTI	0.92	0.88	0.90
LLM-Generated CTI	0.94	0.96	0.95
Hybrid CTI	0.81	0.78	0.79
Overall	0.89	0.87	0.88

5.3. Results and Analysis

Table 1 reports precision, recall, and F1-score across three categories of CTI reports, Authentic, LLM-Generated, and Hybrid, and then an overall performance average. For Authentic CTI, the precision is 0.92, therefore of the CTI reports VeraCTI labelled "Authentic," 92% are actually authentic. The recall is 0.88, meaning of all truly authentic CTI reports, 88% are correctly identified. The F1-Score is 0.90, which reflects high precision and recall, indicating robust performance in detecting legitimate CTI content. For LLM-Generated CTI, the precision is 0.94 and recall is 0.96, leading to an F1-score of 0.95. This near-perfect performance indicates VeraCTI is highly accurate at detecting completely fabricated CTI produced by language models. For Hybrid CTI, precision is 0.81 and recall is 0.78, resulting in an F1-score of 0.79. Hybrid CTI (partially real content with subtle modifications) is more challenging to classify, which is reflected in the comparatively lower F1-score.

VeraCTI's overall precision, recall, and F1-score hover around 0.89, 0.87, and 0.88, respectively. This underscores that the system reliably identifies both fully authentic and fully fabricated reports, and performs moderately well, though not perfectly, on partially manipulated (hybrid) reports.

Examining the misclassifications revealed several patterns. The first is false positives in the authentic CTI Category. Legitimate reports misclassified as false often contained unusual technical details or emerging threats not yet widely documented in external sources, limiting corroboration opportunities. Additionally, reports with minimal IoCs or primarily qualitative intelligence were more likely to be incorrectly flagged, highlighting a potential bias toward IoC-rich intelligence. The second is false negatives in the synthetic CTI category. LLM-generated reports that successfully incorporated verifiable facts and referenced legitimate incidents were occasionally misclassified as authentic, particularly when they maintained internal consistency and avoided verifiable but false claims. The last is the challenges in classifying the hybrid category CTI. The system showed notable difficulty with hybrid reports that preserved most of the original content while subtly altering key technical details or conclusions. This suggests that partial modifications are particularly effective at evading detection, which aligns with findings in related domains such as fake news detection [1].

6. Educational Integration

Beyond improving defensive capabilities in real-world cybersecurity settings, VeraCTI also serves as an educational platform (see Figure 2 and Figure 3). Universities and training programmes can incorporate VeraCTI into hands-on labs, where students learn how to validate and interpret threat reports generated by LLMs. By examining step-by-step probability scoring, novices gain an appreciation for the complexities of CTI credibility assessment, including the significance of IoC corroboration, domain reputation, and internal consistency. Because the tool's reasoning engine presents clear explanations for each flagged inconsistency or match, instructors can use these outputs to highlight best practices in threat hunting and intelligence sharing.

In operational settings, Security Operations Centres (SOCs) can integrate VeraCTI into their threat intelligence workflows as a validation layer for incoming CTI. This serves both to verify threats and train analysts through clear explanations of the verification process. SOC teams can tailor the system by adjusting confidence thresholds and adding specialised data feeds relevant to their security concerns. This practical deployment helps develop analysts' critical thinking skills while strengthening

organisational defences against emerging threats. Moreover, VeraCTI can be configured for automated scanning of incoming threat feeds, reducing analyst workload and accelerating response times for validated threats. Organisations also benefit from reduced operational costs by minimising time spent investigating false positives, while the standardised validation methodology improves communication between incident response, threat hunting, and executive teams who rely on consistent CTI evaluation criteria.

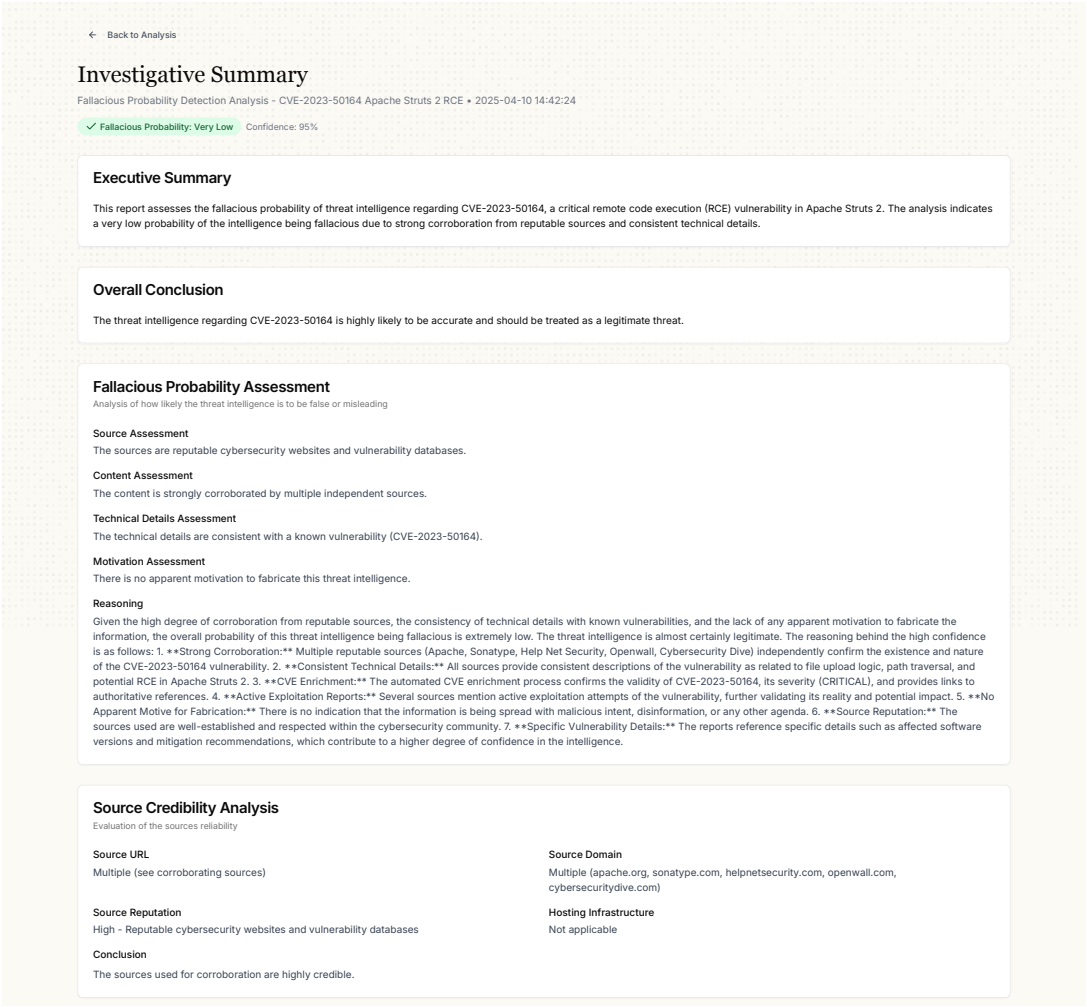

Figure 3: VeraCTI's results page, displaying the confidence score for fallaciousness, source credibility insights, and detailed corroboration analysis for a given CTI report.

7. Conclusion and Future Work

This paper proposed VeraCTI, a framework designed to evaluate the authenticity of CTI reports, especially in light of AI-generated misinformation. By integrating source credibility checks, LLM-based semantic analysis, IoC enrichment, and probability scoring, VeraCTI systematically determines whether a given CTI report is likely to be deceptive or aligned with corroborated intelligence.

From an educational standpoint, VeraCTI offers transparent, step-by-step insights, enabling students, researchers, and security professionals to understand both the benefits and limitations of AI-based CTI validation. Future development will explore domain-specific fine-tuning of the LLMs involved, real-time integration with additional threat intelligence feeds, and further refinements to the probability-scoring methodology. Through these enhancements, we hope to solidify VeraCTI's role as both a powerful validation engine and an educational companion in the constantly evolving field of cybersecurity.

Declaration on Generative AI

The author(s) have not employed any Generative AI tools.

References

[1] M. Hassanin, N. Moustafa, A comprehensive overview of large language models (LLMs) for cyber defences: Opportunities and directions, 2024. arXiv:2405.14487.

[2] Y. Schwartz, L. Ben-Shimol, D. Mimran, Y. Elovici, A. Shabtai, LLMCloudHunter: Harnessing LLMs for automated extraction of detection rules from cloud-based CTI, in: THE WEB CONFERENCE 2025, 2025. URL: https://openreview.net/forum?id=MFUD557wr7.

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

[4] B. C. Das, M. H. Amini, Y. Wu, Security and privacy challenges of large language models: A survey, ACM Comput. Surv. 57 (2025). doi:10.1145/3712001.

[5] E. Iturbe, E. Rios, A. Rego, N. Toledo, Artificial intelligence for next generation cybersecurity: The AI4CYBER framework, in: Proceedings of the 18th International Conference on Availability, Reliability and Security, ARES '23, Association for Computing Machinery, New York, NY, USA, 2023, p. 8. doi:10.1145/3600160.3605051.

[6] P. Ranade, A. Piplai, S. Mittal, A. Joshi, T. Finin, Generating fake cyber threat intelligence using transformer-based models, in: 2021 International Joint Conference on Neural Networks (IJCNN), 2021, pp. 1–9. doi:10.1109/IJCNN52387.2021.9534192.

[7] R. Fayyazi, S. J. Yang, On the uses of large language models to interpret ambiguous cyberattack descriptions, 2023. arXiv:2306.14062.

[8] I. Hasanov, S. Virtanen, A. Hakkala, J. Isoaho, Application of large language models in cybersecurity: A systematic literature review, IEEE Access 12 (2024) 176751–176778. doi:10.1109/ACCESS.2024.3505983.

[9] L. Yang, M. Wang, W. Lou, An automated dynamic quality assessment method for cyber threat intelligence, Computers & Security 148 (2025) 104079. doi:doi.org/10.1016/j.cose.2024.104079.

[10] Z. Wu, F. Tang, M. Zhao, Y. Li, KGV: Integrating large language models with knowledge graphs for cyber threat intelligence credibility assessment, 2024. arXiv:2408.08088.

[11] J. Zhang, H. Bu, H. Wen, Y. Liu, H. Fei, R. Xi, L. Li, Y. Yang, H. Zhu, D. Meng, When LLMs meet cybersecurity: a systematic literature review, Cybersecurity 8 (2025) 55. doi:10.1186/s42400-025-00361-w.

[12] W. Tounsi, H. Rais, A survey on technical threat intelligence in the age of sophisticated cyber attacks, Computers & Security 72 (2018) 212–233. doi:10.1016/j.cose.2017.09.001.

[13] R. Fieblinger, M. T. Alam, N. Rastogi, Actionable cyber threat intelligence using knowledge graphs and large language models, in: 2024 IEEE European Symposium on Security and Privacy Workshops (EuroS&PW), 2024, pp. 100–111. doi:10.1109/EuroSPW61312.2024.00018.

[14] G. Siracusano, D. Sanvito, R. Gonzalez, M. Srinivasan, S. Kamatchi, W. Takahashi, M. Kawakita, T. Kakumaru, R. Bifulco, Time for action: Automated analysis of cyber threat intelligence in the wild, CoRR abs/2307.10214 (2023). doi:10.48550/ARXIV.2307.10214.

[15] Y. Hu, F. Zou, J. Han, X. Sun, Y. Wang, LLM-TIKG: Threat intelligence knowledge graph construction utilizing large language model, Computers & Security 145 (2024) 103999. doi:10.1016/j.cose.2024.103999.

[16] Y. Zhang, T. Du, Y. Ma, X. Wang, Y. Xie, G. Yang, Y. Lu, E.-C. Chang, AttacKG+: Boosting attack graph construction with large language models, Computers & Security 150 (2025) 104220. doi:10.1016/j.cose.2024.104220.

[17] E. Aghaei, X. Niu, W. Shadid, E. Al-Shaer, Securebert: A domain-specific language model

for cybersecurity, in: F. Li, K. Liang, Z. Lin, S. K. Katsikas (Eds.), Security and Privacy in Communication Networks, Springer Nature Switzerland, Cham, 2023, pp. 39–56. doi:10.1007/978-3-031-25538-0_3.

[18] M. Bayer, P. Kuehn, R. Shanehsaz, C. Reuter, Cysecbert: A domain-adapted language model for the cybersecurity domain, ACM Trans. Priv. Secur. 27 (2024). doi:10.1145/3652594.

[19] S. Mitra, S. Neupane, T. Chakraborty, S. Mittal, A. Piplai, M. Gaur, S. Rahimi, LOCALINTEL: Generating organizational threat intelligence from global and local cyber knowledge, 2025. arXiv:2401.10036.

[20] X. Liu, J. Liang, Q. Yan, M. Ye, J. Jia, Z. Xi, Cyber defense reinvented: Large language models as threat intelligence copilots, 2025. arXiv:2502.20791.

[21] Y. Yao, J. Duan, K. Xu, Y. Cai, Z. Sun, Y. Zhang, A survey on large language model (LLM) security and privacy: The good, the bad, and the ugly, High-Confidence Computing 4 (2024) 100211. doi:10.1016/j.hcc.2024.100211.

[22] N. Khurana, S. Mittal, A. Piplai, A. Joshi, Preventing poisoning attacks on ai based threat intelligence systems, in: 2019 IEEE 29th International Workshop on Machine Learning for Signal Processing (MLSP), 2019, pp. 1–6. doi:10.1109/MLSP.2019.8918803.

[23] Z. Song, Y. Tian, J. Zhang, Y. Hao, Generating Fake Cyber Threat Intelligence Using the GPT-Neo Model, in: 2023 8th International Conference on Intelligent Computing and Signal Processing (ICSP), IEEE Computer Society, Los Alamitos, CA, USA, 2023, pp. 920–924. doi:10.1109/ICSP58490.2023.10248596.

[24] A. Tundis, S. Ruppert, M. Mühlhäuser, On the automated assessment of open-source cyber threat intelligence sources, in: V. V. Krzhizhanovskaya, G. Závodszky, M. H. Lees, J. J. Dongarra, P. M. A. Sloot, S. Brissos, J. Teixeira (Eds.), Computational Science – ICCS 2020, Springer International Publishing, Cham, 2020, pp. 453–467. doi:10.1007/978-3-030-50417-5_34.