DFIR-Metric: A Benchmark Dataset for Evaluating Large Language Models in Digital Forensics and Incident Response

Bilel Cherif[1], Tamas Bisztray[2], Richard A. Dubniczky[3], Aaesha Aldahmani[1], Saeed Alshehhi[1], and Norbert Tihanyi[1,3(✉)]

[1] Technology Innovation Institute, Abu Dhabi, UAE
{bilel.cherif,saeed.alshehhi,aaesha.aldahmani,norbert.tihanyi}@tii.ae
[2] University of Oslo, Oslo, Norway
tamasbi@ifi.uio.no
[3] Eötvös Loránd University, Budapest, Hungary
{dubniczky,ntihanyi}@inf.elte.hu

Abstract. Digital Forensics and Incident Response (DFIR) involves analyzing digital evidence to support legal investigations. Large Language Models (LLMs) offer new opportunities in DFIR tasks such as log analysis and memory forensics, but their susceptibility to errors and hallucinations raises concerns in high-stakes contexts. Despite growing interest, there is no comprehensive benchmark to evaluate LLMs across both theoretical and practical DFIR domains. To address this gap, we present DFIR-Metric, a benchmark with three components: (1) Knowledge Assessment: a set of 700 expert-reviewed multiple-choice questions sourced from industry-standard certifications and official documentation; (2) Realistic Forensic Challenges: 150 CTF-style tasks testing multi-step reasoning and evidence correlation; and (3) Practical Analysis: 500 disk and memory forensics cases from the NIST Computer Forensics Tool Testing Program (CFTT). We evaluated 14 LLMs using DFIR-Metric, analyzing both their accuracy and consistency across trials. We also introduce a new metric, the Task Understanding Score (TUS), designed to more effectively evaluate models in scenarios where they achieve near-zero accuracy. This benchmark offers a rigorous, reproducible foundation for advancing AI in digital forensics. All scripts, artifacts, and results are available on the project website at https://github.com/DFIR-Metric.

Keywords: Digital Forensics · Incident Response · LLM Benchmarking

1 Introduction

Since the Turing Test first challenged machines to mimic human conversation [32], progress in *Natural Language Processing (NLP)* has been tracked through various benchmarks. As noted by Wang et al. [35], modern *Large Language Models (LLMs)*, powered by neural networks and transformers [33], often record near-perfect scores on widely used suites such as GLUE and SQuAD [34,23], which

reduces the effectiveness of these tests. In response, some new benchmarks like FrontierMath [9] are made future-proof, and even advanced models can only achieve 1.7% accuracy. These highly complex benchmarks do not support clear differentiation between the capabilities of current models. LLMs hold immense potential for various fields, including cybersecurity [31], software engineering [22], biomedicine [4] or law [6], which has sparked calls for privacy-aware, reliability-oriented, and domain-tailored benchmarks [35].

Digital Forensics and Incident Response (DFIR) is one such domain where practitioners analyze logs, e-mails, and multilingual reports to identify evidence, reconstruct timelines, and mitigate threats [11]. Recent studies show promising results when LLMs are applied in the DFIR domain, particularly for log filtering, artifact classification, and incident reporting [25,38,17,16,21]. However, the stakes are especially high. Errors can compromise evidence or misdirect investigations, and the use of proprietary models may violate strict confidentiality requirements. LLMs are known to hallucinate facts and misinterpret context [28]. Before they can be trusted in DFIR workflows, we need rigorous, task-specific evaluations that measure not only one-off success through accuracy but also reliability and consistency.

Evaluating the performance of LLMs within the DFIR domain remains a significant challenge due to the absence of a comprehensive benchmark datasets and well-defined evaluation metrics. Although several strong general-purpose and domain-specific benchmarks are available, none provide a comprehensive evaluation across the diverse landscape of DFIR. As a result, practitioners lack a clear framework to determine when LLMs can be reliably applied and when expert validation is still required. A question naturally rises: "*Which specific DFIR tasks can LLMs effectively support, and in which areas is human expertise still essential?*" To obtain a detailed answer, we frame the study around the following research questions:

Research Questions

RQ1: What level of comprehension and confidence do LLMs exhibit in DFIR domain knowledge when challenged with certification-grade multiple-choice questions?
RQ2: To what extent can LLMs accurately and reliably solve practical forensic workflows, like log triage, memory-dump analysis, reverse engineering, and string search?
RQ3: Among the leading proprietary models and the strongest open-source alternatives, which achieve the highest scores in a unified evaluation?

To the best of our knowledge, no comprehensive and standardized benchmark currently exists in the literature to thoroughly address these research questions. To fill this gap, we introduce `DFIR-Metric`, a novel suite of benchmark tasks and datasets to evaluate LLMs in the DFIR domain. According to NIST Special Publication 800-86 *"Guide to Integrating Forensic Techniques into Incident Response"* [13], the digital forensics process consists of five key steps: identifying evidence, collecting artifacts, examining data, analyzing findings, and reporting

results. Our benchmark evaluates LLMs on the first four stages, emphasizing technical accuracy and procedural rigor, while intentionally excluding the final legal reporting phase. This paper makes the following three key contributions:

- **DFIR-Metric**: We design a three-part dataset to evaluate LLMs on: (a) DFIR knowledge, using 700 human-verified multiple-choice questions sourced from industry certifications and official documentation; (b) practical disk and memory forensics tasks, evaluated through the string search tests of NIST's *Computer Forensics Tool Testing Program* (CFTT); and (c) CTF-style challenges on realistic forensic investigations that require planning, analytical reasoning, and evidence correlation.
- **Improved Evaluation Metrics**: Beyond single-pass accuracy, we assess each task multiple times to ensure reliability and introduce a new evaluation metric: the *Task Understanding Score* (TUS), which rewards models for accurately completing steps in a multi-step pipeline;
- **Reproducability**: All artifacts, associated scripts, and the final `DFIR-Metric` dataset are available on the project's GitHub page, allowing independent researchers to integrate new models, replicate our results, and expand the evaluation as needed. We assessed 14 state-of-the-art LLMs to capture the current landscape of model advancements. (https://github.com/DFIR-Metric)

The remainder of the paper is organized as follows. Section 2 reviews related work. Section 3 outlines the methodology used to construct the benchmark's three components. Section 4 presents experimental results for various state-of-the-art LLMs, while Section 5 concludes the paper.

Ethical Considerations

All 700 DFIR-Metric questions were built from publicly available sources. Any text that resembled certification material was paraphrased or abstracted to prevent direct association with specific certification bodies or copyrighted material, and brief quotations are used only for research—a context generally covered by fair-use (or equivalent) provisions. The benchmark is independent of, and unendorsed by any certification body. Our aim is to support open, ethical research while respecting the rights of certification providers, content creators, and the broader digital forensics community.

2 Related Work

Ferrag et. al [7] identified nine main areas where LLMs are being used today marking DF as a standalone field. Sharma et al. [26] proposed ForensicLLM, a model fine-tuned on a custom Q&A dataset for digital forensics tasks, but neither the model nor its dataset is publicly released. Several recent studies have explored the use of pre-trained large language models for a variety of tasks across the digital forensic investigation pipeline. These tasks include timeline reconstruction [16], automated report writing [18], and technical analyses such as malware detection and reverse engineering [8,12]. Other work has focused on artifact examination [24], as well as more practical applications like evidence

extraction, scripting automation, and data recovery [24]. These studies highlight the growing interest in adapting LLMs to support various stages of forensic workflows, though many remain exploratory in nature. Wickramasekara et al. [37] introduced AutoDFBench to assess AI coding skills in string search using NIST CFTT test suites. We note that Module III of `DFIR-Metric`, which will be detailed in the Section 3, also utilizes the NIST CFTT challenges to assess LLMs' capabilities in string search tasks.

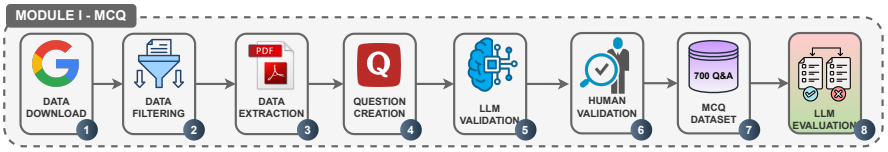
Most existing evaluation frameworks for LLM-based forensics still rely on ad-hoc, chat-style prompt tests. This approach does not scale well to large question sets, provides limited control over response variability, and poses significant challenges for reproducibility. Horsman and Lyle [10] emphasized the lack of high-quality datasets in digital forensics and proposed several guiding principles for dataset creation. Expanding on their work, we identify four key requirements that any robust forensic benchmark should meet: (i) publicly accessible, well-organized benchmarks hosted on platforms such as GitHub or Hugging Face; (ii) a plug-and-play evaluation framework that allows models to be tested via simple API integration; (iii) evaluation metrics that go beyond accuracy to include critical risks such as hallucination frequency and domain-specific blind spots; and (iv) a formal dataset specification that adopts a standardized format to support auditability and long-term reproducibility. While such datasets exist in broader cybersecurity domains, for example [30], none of the existing DFIR-specific datasets fully meet all the criteria outlined above. Table 1 lists datasets, benchmarks, and frameworks that satisfy a subset of the criteria and support the DFIR field.

Table 1: DFIR-Relevant Datasets and Benchmarks

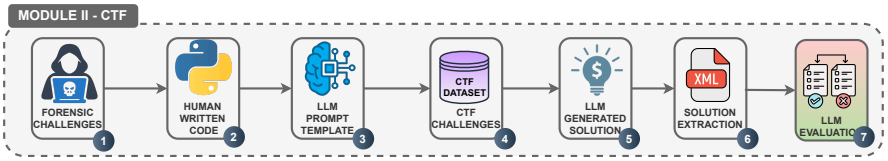
Dataset Name	**Modality**	**Benchmark Scope / Description**
`CyberMetric` [31]	Textual (MCQ)	Cybersecurity benchmark with 10,000 MCQ;
`DIA-Bench` [30]	Textual (JSON)	Cybersewcurity/math reasoning benchmark
`RAISE` [5]	Multimedia (Images)	Camera-native images to support classification tasks
`Vision Forensics` [27]	Multimedia (Video)	Device-attributed video samples for integrity analysis
`Timeline Analysis` [29]	Timeline (CSV/JSON)	Plaso-based timeline QA benchmark for evaluating LLMs
IoT-CAD [19]	Memory, Disk, Net	Labeled IoT attack traces with memory and network data.
`DeepSpeak` [2]	Multimedia (Aud/Vid)	100h webcam speech for deepfake detection tasks
`CIC-MalMem-2022` [3]	Memory Dumps	58k labeled Windows dumps (malware/benign)
`Unraveled` [20]	Logs (Net + Host)	Multi-week APT simulation with labeled detection logs.
`SCVIC-APT-2021` [14]	Network (pcap)	APT emulation with attack phases and labeled flows
`SCVIC-CIDS-2021` [15]	Logs (Net + Host)	Host/network logs combining CIC-IDS-2018 traffic traces
`AutoDFBench` [36]	Disk + Artifacts	AI tool validation benchmark against NIST CFTT
`CTIBench` [1]	Textual (TSV)	CTI benchmark with threat classification, CVSS scoring

3 Methodology: Dataset Creation and Evaluation Metrics

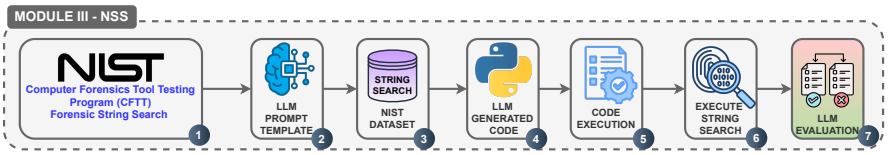
`DFIR-Metric` consists of three core components, as shown in Figure 1. Module I focuses on 700 multiple-choice question generation and evaluation. Module II contains CTF-style forensic challenges, covering a wide range of real-world scenarios from comprehensive log analysis to reverse engineering tasks. Module III introduces the NIST CFTT String Search Challenge[1], requiring LLMs to apply advanced forensic skills to analyze disk images and locate specific artifacts.



(a) **Module I:** Multiple-Choice Question Dataset Generation and LLM Evaluation



(b) **Module II:** CTF-style Challenges Dataset Generation and LLM Evaluation



(c) **Module III:** NIST String Search Dataset Generation and LLM Evaluation

Fig. 1: DFIR-Metric evaluation framework, consisting of three modules.

3.1 Module I - Multiple-Choice Questions (Static)

To assess theoretical competence in DFIR, we built a high-quality multiple-choice dataset aligned with international standards and certifications. An eight-step pipeline (Figure 1a.) harvested candidate questions from peer-reviewed articles, official guidelines, and certification exams, followed by an LLM grammar check and a 200-hour expert review. Ambiguous questions such as "'*Where are deleted files stored in Windows operating systems?*" were revised to eliminate imprecision. In Windows 10, deleted files reside in `C:\$Recycle.Bin`, whereas

[1] https://www.nist.gov/itl/ssd/software-quality-group/computer-forensics-tool-testing-program-cftt

in Windows XP, they are located in `C:\RECYCLER`. This module tests only knowledge through multiple-choice questions and does not assess the practical skills required to perform forensic tasks. Practical skillsets will be evaluated in the following modules. An example question is shown in the code snippet below:

> `Question example (Module 1):` Which command can provide the investigators with details of all the loaded modules on a Linux-based system?
> **A:** "plist mod -a", **B:** "lsof -m", **C:** "lsmod", **D:** "list modules -a".

3.2 Module II - CTF-style Forensic Challenges (Dynamic)

Inspired by *Capture-the-Flag* (CTF) events, this module evaluates log analysis, cryptographic puzzles, and system-forensics skills. This is a dynamic module where each task is based on a hand-crafted template. Parameters such as log lines, keys, file system artifacts, and attacker actions can be randomized to generate multiple unique instances of the same task. In the evaluation we probe each task template three times to test the reliability of LLMs in solving specific tasks. Figure 1b. outlines the pipeline. All templates and solutions were manually audited, preserving real-world DFIR complexity while providing a controlled ground truth for rigorous, reasoning-centric assessment. Some of the CTF templates are modified versions from our previous work [30], while several brand new task were added for forensics. An example question is shown below:

> `Question example (Module 2):` Find the flag in this hex dump. Note: Characters are XOR'ed with 0x55 before hex encoding **0x0000: 3f d7 8c 31 78 e0 4d 00 4d 3b fb 69 71 66 9a 26 0x0010: 99 0f f3 a6 16 21 9b a5 82 36 5a 90 28** .....

3.3 Module III - NIST Forensic String Search (Static)

The third module introduces hands-on disk analysis tasks focused on string search, a fundamental forensic technique. This benchmark is based on the *NIST Computer Forensics Tool Testing Program*'s technical documentation, originally designed to evaluate tools like EnCase and Magnet AXIOM using standardized datasets such as the String Search Test Data Set Package Version 1.1, which contains known content across various file systems. To adapt these challenges for LLMs, we reformulated each task into a prompt accompanied by a valid disk image, asking the model to generate a Python script to solve the given forensic problem. To assess performance, we developed an automated evaluation pipeline that analyzes disk images, extracts memory blocks, parses file systems, and recovers both active and deleted files. This output was used to construct ground truth baselines, which were rigorously validated by human experts. These baselines served as reference outputs for evaluating and comparing LLM-generated responses across tasks. The entire process is illustrated in Figure 1c.

3.4 Defining Task Understanding Score for LLM evaluation

In [30] four novel metrics were introduced; *Reliability Score* (RS@k), *Task Success Rate* (TSR@k), *Confidence Index* (Conf@k), and *Near Miss Score* (NMS@k). Let t represent a question template with variable parameters, and let $\mathcal{T} = \{t_1, t_2, \ldots, t_n\}$ denote the set of all such templates. Let $\mathcal{Q}(\mathcal{T}, k) = \{q_1, q_2, \ldots, q_{n\times k}\}$ denote the set of unique questions, where each template from $\mathcal{T}$ is used to generate k different questions and let $\mathcal{S}_\mathcal{Q} = \{s_1, s_2, \ldots, s_{n\times k}\}$ represent the set of solutions corresponding to $\mathcal{Q}$. We have $f : \mathcal{Q}(\mathcal{T}, k) \to \mathcal{S}_\mathcal{Q}$ such that $f(q_i) = s_i$ for all $i \in \{1, \ldots, n \times k\}$.

Definition 1 (Reliability Score). *The* Reliability Score (RS@k) *over a dataset* $\mathcal{Q}(\mathcal{T}, k)$ *is calculated as:*

$$RS@(k) = \frac{1}{k} \sum_{i=1}^{n\times k} \mathcal{A}_i \tag{1}$$

where $\mathcal{A}_i$ *denotes the score assigned to answering* q_i*, defined as* $\mathcal{A}_i = +1$ *if* s_i *is correctly returned for* q_i*,* 0 *if* q_i *is skipped, and* -2 *otherwise.*

Definition 2 (Task Success Rate). *The* Task Success Rate (TSR@(t_i, k)) *evaluates the number of correct answers for a given question template* t_i *out of the* k *generated instances, where the number of templates is* $i \in \{1, 2, \ldots, n\}$*.*

$$TSR@(t_i, k) = \sum_{j=1}^{k} \mathcal{B}_j \tag{2}$$

where the value of $\mathcal{B}_j$ *is defined as* $\mathcal{B}_j = +1$ *if* s_j *is returned for* q_j*,* 0 *otherwise.*

Definition 3 (Confidence Index). *The* Confidence Index (Conf@k) *represents the percentage of question templates in a dataset where, for a given template* t_i*, all* k *generated queries are successfully answered,*

$$Conf@(k) = \frac{100}{n} \sum_{i=1}^{n} \begin{cases} 1 & \text{if } TSR@(t_i, k) = k \\ 0 & \text{otherwise.} \end{cases} \tag{3}$$

A new metric - The Task Understanding Score (TUS) Metrics such as TSR@k, Conf@k, and the traditional Pass@k assess whether a response to question q_i is fully correct, but they do not account for cases where an LLM demonstrates partial success on a task. If LLMs score zero on a given task, it cannot establish a meaningful ranking, nor it will provide insight on how close models are to the correct solution. In reality, answers frequently contain some correct components, and we should also give credit for partial correctness. We want to move beyond simply classifying answers as correct or not, and introduce more granular scoring. Let $\mathcal{C} = \{c_1, c_2, \ldots, c_m\}$ be the set of key criteria, where $m = |\mathcal{C}|$. Each criterion can represent various aspects—for instance, whether the Python code generated by an LLM executes correctly, or whether key steps

essential to solving the problem are present. We can measure how many criteria are satisfied when solving question q_i. Let $r_{ij} \in \{0, 1\}$ indicate whether the j-th criterion is satisfied in the solution to question q_i. Then:

Definition 4 (Task Understanding Score). *The* Task Understanding Score (TUS@m) *quantifies how well responses capture the essential components of a solution. It measures the average proportion of criteria satisfied across all evaluated responses.*

$$TUS@m = \frac{1}{|\mathcal{Q}|} \sum_{i=1}^{|\mathcal{Q}|} \left(\frac{1}{m} \sum_{j=1}^{m} r_{ij} \right) \tag{4}$$

where $|Q|$ is the total number of evaluated questions.

Using TUS@m, we can evaluate the performance of LLMs on challenging tasks where traditional metrics like accuracy often yield a score of zero. Even in such cases, TUS@m enables us to capture partial correctness by assessing which predefined building block of a solution is satisfied in the response. For Module III tasks, the number of criteria $|\mathcal{C}|$ is set to four ($m = 4$), with a dataset comprising $|\mathcal{Q}| = 500$ NIST Forensic String Search challenge.

4 Experimental results

To asses the capabilities of different LLMs on the newly introduced `DFIR-Metric` we conducted an experiment on multiple commercial and open-sourced models.

4.1 Module I - Multiple-Choice Questions

We evaluated 14 state-of-the-art models on the MCQ dataset. Each question was asked 3 times, where correct answer was randomized between A, B, C or D to eliminate guessing. The best-performing model was GPT-4.1, closely followed by GPT-4o and Grok 3 with only marginal differences. Among the open-source, non-proprietary models, the best performer was Qwen-2.5 with 72 billion parameters. It achieved a Confidence Index (CI) of 84.29% with $k = 3$ and a Mean Accuracy (MA) of 89.90%, which is only 5% lower than the state-of-the-art GPT models. Table 2 displays the final results of the 14 LLMs tested.

4.2 Module II - CTF-style Forensic Challenges

Each CTF task was issued as a single prompt. Following Definition 1, models earned +1 for a correct response, 0 for skip, and −2 for an incorrect answer. All prompts, tasks, and the Google Colab code are published on our GitHub page to support easy and reproducible research. The evaluations were conducted via API, and no code execution was performed by the models in this module—consistent with their standard API capabilities. This contrasts with Module III, where Python code execution was preformed for the NIST string search tasks. Table 3 presents the final results of the tested LLMs on the CTF challenges.

Table 2: LLM Performance on 700 MCQ dataset (k=3) (Sorted by CI)

Model	Company	Size	License	Open	CI	MA
GPT-4.1	OpenAI	N/A	Proprietary	✘	89.34%	92.75%
GPT-4o	OpenAI	N/A	Proprietary	✘	88.92%	93.03%
Grok 3	xAI	2700B	Proprietary	✘	88.22%	91.91%
Claude 3.7 Sonnet	Anthropic	N/A	Proprietary	✘	86.40%	91.58%
Gemini 2.5 Flash	Google	N/A	Proprietary	✘	85.41%	90.37%
Qwen-2.5	Qwen	72B	Apache 2.0	✔	84.29%	89.80%
DeepSeek V3	DeepSeek AI	671B	DeepSeek	✔	81.76%	89.25%
GPT-4o-mini	OpenAI	N/A	Proprietary	✘	79.94%	85.78%
Llama 3.3	Meta	70B	Llama 3	✔	79.80%	86.49%
WizardLM 2	Microsoft	8×22B	Apache 2.0	✔	77.84%	84.53%
Gemma 3	Google	27B	Gemma	✔	77.13%	84.71%
Mixtral-8x7B	Mistral AI	46.7B	Apache 2.0	✔	71.11%	80.36%
Gemma 2	Google	9B	Gemma	✔	68.58%	79.85%
Mistral-3B	Mistral AI	3B	Apache 2.0	✔	25.66%	55.86%

Table 3: DFIR-Metric CTF Performance ($k = 3$) (Sorted by Confidence Index)

Model	Company	Size	Open	Correct	Skipped	Wrong	RS	CI
GPT-4.1	OpenAI	N/A	✘	47	0	103	-53.0	28%
GPT-4o	OpenAI	N/A	✘	46	18	86	-42.0	26%
DeepSeek V3	DeepSeek AI	671B	✔	43	18	89	-45.0	22%
Qwen-2.5	Qwen	72B	✔	35	14	101	-55.7	20%
Llama3.3	Meta	70B	✔	33	16	101	-56.3	20%
Grok 3	xAI	2700B	✘	40	5	105	-56.7	20%
Gemini 2.5-flash	Google	N/A	✘	38	1	111	-61.3	20%
GPT-4o-mini	OpenAI	N/A	✘	27	14	109	-63.7	20%
Claude 3.7 Sonnet	Anthropic	N/A	✘	18	0	132	-82.0	12%
Mixtral-8x7B	Mistral AI	47B	✔	24	1	125	-75.3	12%
Gemma 3	Google	27B	✔	22	2	126	-76.7	10%
Mistral	Mistral AI	3B	✔	22	2	126	-76.7	10%
Gemma 2	Google	9B	✔	0	0	150	-100.0	0%

Table 4: NIST Forensic String Search ($m = 4$) (Sorted by TUS@4)

Model	Company	Size	Open	Correct	Syntax	Wrong	T/O	TUS@4
GPT-4.1	OpenAI	N/A	✘	1	217	292	0	38.52%
GPT-4o	OpenAI	N/A	✘	0	283	226	1	27.99%
Gemini 2.5-flash	Google	N/A	✘	1	16	493	0	25.88%
Claude 3.7 Sonnet	Anthropic	N/A	✘	5	309	195	1	24.75%
DeepSeek V3	DeepSeek AI	671B	✔	0	268	13	229	22.40%
Grok 3	xAI	2700B	✘	0	316	194	0	21.71%
Llama3.3	Meta	70B	✔	0	293	7	210	15.40%
GPT-4o-mini	OpenAI	N/A	✘	0	51	459	0	12.64%
Qwen-2.5	Qwen	72B	✔	0	453	56	1	3.62%

GPT-4.1 achieved the highest *confidence index* (CI, 28 %), but its reliability score (RS) was lower than that of GPT-4o and DeepSeek V3 because it attempted every task and accumulated 103 wrong answers, which highlights a potential architectural difference or system prompt design. The other GPT-4 variant skipped 18 uncertain items, incurring fewer penalties and posting the best reliability score (−42).

Among the non-proprietary, openly released models, DeepSeek V3, Qwen-2.5, and Llama-3.3 form a tight cluster (CI 22–20 %), trailing GPT-4.1 by only 6–8 percentage points. Note, that the benchmark includes high-difficulty items—reverse engineering, cryptographic analysis, and large-scale log triage—that typically require code execution or multi-step reasoning. Although larger or more sophisticated models may one day solve these problems in a single pass, present systems should at minimum recognize their limits and abstain, rather than emit fluent yet incorrect answers. The frequency of such errors drives every model's reliability score (range −300 to 150) into negative territory. In this evaluation, partial success was not graded, however, in some cases models were able to extract the correct answer, but failed to return it in the requested format.

4.3 Module III - NIST Forensic String Search

We assessed the nine highest-scoring models from the MCQ benchmark on the NIST Forensic String Search task. The following task serves as an example; "*You are an experienced Digital Forensics and Incident response specialist. Answer the following question by writing a python script.....* Additionally, each prompt contained specific instruction like directing the LLMs to produce a self-contained Python script, and return its findings in a rigid JSON-like list of `<inode>:<filename>` pairs, prefixed by `DELETED` or `LIVE`.

This setting stresses both technical competence (correct identification of strings in a forensic image) and compliance with a brittle output specification—two dimensions that everyday DFIR workflows routinely demand. To calcualte TUS@4, partial points can be awarded for the following categories: **(1)** determining the right offset of the file system from the prompt description, as each image contains 3 different file systems. **(2)** properly identify the image path in the directory. **(3)** identify the correct search string target, and if it requires regex or regular search. **(4)** identifying the right extension for the artefact; doxc, txt, html, etc. Table 4 provides insights on how many occasions models were able to solve the task, or if they failed what was the main reason. The categories are: **Correct:** the script successfully extracts the target information from the forensic. **Wrong:** the script runs but fails to extract the correct datadisk image. **Timeout Execution:** the script does not complete within a predefined execution time. **Syntax Error:** the script fails to run due to code syntax issues.

Although GPT-4.1 secures the highest TUS@4 (38.5 %), its advantage stems largely from a higher rate of *partially* correct steps, rather than from wholesale task completion. Manual review revealed three recurrent error patterns across models: they sometimes hallucinate files, bash commands, paths or libraries that are absent from the image, causing the script to crash; even when the search

logic is sound, the script may capture the wrong sub-string or omit a required field, producing only partially valid lines, and finally; tiny deviations from the rigid output schema, misplaced brackets, missing prefixes or commas invalidate otherwise correct answers.

Table 4 reports the results for the evaluation. TUS rewards incremental progress (correct code fragments, partially valid lists, etc.) rather than binary success. From a practitioner's standpoint this nuance matters: a higher TUS model may still require substantial fine-tuning to yield admissible, reproducible evidence, but it has a step in the right direction. Finally, we note that open-weight models (e.g. DeepSeek V3, Qwen-2.5, Llama 3.3) have yet to match the proprietary leaders in this task.

5 Conclusion

In this work, we introduce `DFIR-Metric`, the first extensive benchmark tailored to evaluate both theoretical knowledge and practical proficiency of LLMs in the domain of Digital Forensics & Incident Response (DFIR). Spanning 700 certification-grade multiple-choice questions, a NIST-compliant string-search suite, and dynamically generated CTF investigations, `DFIR-Metric` evaluates models across the first four phases of the NIST 800-86 forensic workflow. To measure not only accuracy but also consistency and self-assessment, we used reliability metrics such as Confidence Index and Reliability Score for the evaluation, and introduced a novel metric—the Task Understanding Score (TUS)— and executed every task multiple times for awarding partial task completion. Our work addressed three research questions:

- **RQ1**: *What level of comprehension and confidence do LLMs exhibit in DFIR domain knowledge when challenged with certification-grade multiple-choice questions?*
  Answer: The leading models demonstrate substantial mastery of core DFIR principles. GPT-4.1 achieves a Confidence Index of 89.34% and a Mean Accuracy of 92.75%. This underlines that high accuracy does not correspond with reliable problem solving, as models may guess and provide a correct answer by chance. This highlights the importance of repetitive testing and reliability metrics. The open-source Qwen-2.5-72B trails by only 5%, indicating a narrowing proprietary edge, whereas compact models (e.g., Mistral-3B) perform scarcely above pure chance.
- **RQ2**: *To what extent can LLMs accurately and reliably solve practical forensic workflows such as log triage, memory-dump analysis, reverse engineering, and string search?*
  Answer: Practical competence lags behind domain knowledge. In the NIST String Search module, no model produced meaningful results across the 500 prompts, and even the top performer (GPT-4.1) achieved just 38% partial-credit on the Task Understating Score, indicating incomplete pipeline execution (e.g., script generation succeeded but filesystem carving failed). In our CTF-style trials, the best model was again GPT-4.1, but solved only 28% of tasks consistently. Notably, unlike other top performing models like GPT-4o, DeepSeek V3, or Qwen-2.5, GPT-4.1 was not able to skip any questions, highlighting severe limitations in comprehension and self reflection.

– **RQ3**: *Among the leading proprietary models and the strongest open-source alternatives, which achieve the highest scores in a unified evaluation?*
Answer: Overall, the proprietary models, GPT-4.1 and GPT-4o consistently lead in every Module: domain knowledge, CTF challenges, and NIST sting search tasks (although in the latter they were not able to solve a single task, and only achieve partial success through the task understanding score). Among the open source models Qwen–2.5 and DeepSeek V3 perform best in the multiple choice questions, Llama 3.3, WizardLM 2 and Gemma 3 are not trailing far behind. Interestingly, in the CTF challenges DeepSeek V3 performs very close to GPT-4o, skipping the same amount of questions and only getting a 4% worse Confidence Index.

Our findings highlight steady progress but also underscore unresolved challenges in automating end-to-end DFIR investigations. Current LLMs can recall certification material and generate competent forensic scripts, yet struggle with sustained deductive reasoning, rigorous chain-of-custody logic, and calibrated confidence. Here it is important to highlight that we did not include reasoning models in the evaluation like o4-mini or DeepSeek R1, where we expect these models to perform slightly better based on [30].

DFIR-Metric fills a critical evaluation gap, offering the community an open, extensible framework to measure future advances. We release all datasets, grading code, and baseline results to foster reproducibility and encourage iterative enhancement. We conclude that practical digital forensic scenarios and end-to-end forensic workflows remain out of reach for current models.

Acknowledgments. This research is supported and funded by the Technology Innovation Institute (TII), Abu Dhabi. Additional support is provided by ZEISS Digital Innovation; TKP2021-NVA Funding Scheme under Project TKP2021-NVA-29; ELTE-OTP Cyberlab—a collaboration between Eötvös Loránd University (ELTE) and OTP Bank Plc; EPSRC grant EP/T026995/1 titled "EnnCore: End-to-End Conceptual Guarding of Neural Architectures" under the Security for All in an AI-enabled Society program; the Research Council of Norway Project No. 312122 "Raksha: 5G Security for Critical Communications"; funding from Horizon Europe under Grant Agreement No. 101120853; and funding under Grant Agreement No. 101145874, supported by the European Cybersecurity Competence Centre.

Disclosure of Interests. Competing Interests: The authors declare no competing interests relevant to the content of this article.

References

1. Alam, M.T., Bhusal, D., Nguyen, L., Rastogi, N.: Ctibench: A benchmark for evaluating llms in cyber threat intelligence. In: Advances in Neural Information Processing Systems 37 (NeurIPS 2024), Datasets and Benchmarks Track (2024)
2. Barrington, S., Bohacek, M., Farid, H.: The DeepSpeak Dataset (Apr 2025). https://doi.org/10.48550/arXiv.2408.05366, arXiv:2408.05366 [cs] version: 3
3. Carrier, T., Victor, P., Tekeoglu, A., Lashkari, A.: Detecting obfuscated malware using memory feature engineering. In: Proceedings of the 8th International Conference on Information Systems Security and Privacy. SCITEPRESS - Science and Technology Publications (2022)

4. Chen, Q., Hu, Y., Peng, X., Xie, Q., Jin, Q., Gilson, A., Singer, M.B., Ai, X., Lai, P.T., Wang, Z., Keloth, V.K., Raja, K., Huang, J., He, H., Lin, F., Du, J., Zhang, R., Zheng, W.J., Adelman, R.A., Lu, Z., Xu, H.: Benchmarking large language models for biomedical natural language processing applications and recommendations. Nat. Commun. **16**(1), 3280 (Apr 2025)
5. Dang-Nguyen, D.T., Pasquini, C., Conotter, V., Boato, G.: RAISE: a raw images dataset for digital image forensics. In: Proceedings of the 6th ACM Multimedia Systems Conference. pp. 219–224. MMSys '15, Association for Computing Machinery, New York, NY, USA (Mar 2015). https://doi.org/10.1145/2713168.2713194
6. Fei, Z., Shen, X., Zhu, D., Zhou, F., Han, Z., Huang, A., Zhang, S., Chen, K., Yin, Z., Shen, Z., Ge, J., Ng, V.: LawBench: Benchmarking legal knowledge of large language models. In: Al-Onaizan, Y., Bansal, M., Chen, Y.N. (eds.) Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing. pp. 7933–7962. Association for Computational Linguistics, Miami, Florida, USA (Nov 2024). https://doi.org/10.18653/v1/2024.emnlp-main.452
7. Ferrag, M.A., Alwahedi, F., Battah, A., Cherif, B., Mechri, A., Tihanyi, N., Bisztray, T., Debbah, M.: Generative AI in cybersecurity: A comprehensive review of LLM applications and vulnerabilities. Internet of Things and Cyber-Physical Systems **5**, 1–46 (2025). https://doi.org/10.1016/j.iotcps.2025.01.001
8. Ferrag, M.A., Ndhlovu, M., Tihanyi, N., Cordeiro, L.C., Debbah, M., Lestable, T., Thandi, N.S.: Revolutionizing Cyber Threat Detection With Large Language Models: A Privacy-Preserving BERT-Based Lightweight Model for IoT/IIoT Devices. IEEE Access **12**, 23733–23750 (2024). https://doi.org/10.1109/ACCESS.2024.3363469
9. Glazer, E., Erdil, E., Besiroglu, T., Chicharro, D., Chen, E., Gunning, A., Olsson, C.F., Denain, J.S., Ho, A., Santos, E.d.O., Järviniemi, O., Barnett, M., Sandler, R., Vrzala, M., Sevilla, J., Ren, Q., Pratt, E., Levine, L., Barkley, G., Stewart, N., Grechuk, B., Grechuk, T., Enugandla, S.V., Wildon, M.: FrontierMath: A Benchmark for Evaluating Advanced Mathematical Reasoning in AI (Dec 2024). https://doi.org/10.48550/arXiv.2411.04872, http://arxiv.org/abs/2411.04872, arXiv:2411.04872 [cs]
10. Horsman, G., Lyle, J.R.: Dataset construction challenges for digital forensics. Forensic Science International: Digital Investigation **38**, 301264 (Sep 2021). https://doi.org/10.1016/j.fsidi.2021.301264
11. Johansen, G.: Digital Forensics and Incident Response. Packt Publishing, Birmingham, England, 2 edn. (Jan 2020)
12. Joyce, R.J., Patel, T., Nicholas, C., Raff, E.: AVScan2Vec: Feature Learning on Antivirus Scan Data for Production-Scale Malware Corpora. In: Proceedings of the 16th ACM Workshop on Artificial Intelligence and Security. pp. 185–196. AISec '23, Association for Computing Machinery, New York, NY, USA (Nov 2023). https://doi.org/10.1145/3605764.3623907
13. Kent, K., Chevalier, S., Grance, T., Dang, H.: Guide to Integrating Forensic Techniques into Incident Response. Tech. Rep. NIST Special Publication (SP) 800-86, National Institute of Standards and Technology (Sep 2006). https://doi.org/10.6028/NIST.SP.800-86
14. Liu, J., Shen, Y., Simsek, M., Kantarci, B., Mouftah, H.T., Bagheri, M., Djukic, P.: A New Realistic Benchmark for Advanced Persistent Threats in Network Traffic. IEEE Networking Letters **4**(3), 162–166 (Sep 2022). https://doi.org/10.1109/LNET.2022.3185553

15. Liu, J., Simsek, M., Kantarci, B., Bagheri, M., Djukic, P.: Collaborative Feature Maps of Networks and Hosts for AI-driven Intrusion Detection. In: GLOBECOM 2022 - 2022 IEEE Global Communications Conference. pp. 2662–2667 (Dec 2022). https://doi.org/10.1109/GLOBECOM48099.2022.10000985, iSSN: 2576-6813
16. Loumachi, F.Y., Ghanem, M.C., Ferrag, M.A.: Advancing Cyber Incident Timeline Analysis Through Retrieval-Augmented Generation and Large Language Models. Computers **14**(2), 67 (Feb 2025). https://doi.org/10.3390/computers14020067, number: 2 Publisher: Multidisciplinary Digital Publishing Institute
17. Michelet, G., Breitinger, F.: Chatgpt, llama, can you write my report? an experiment on assisted digital forensics reports written using (local) large language models. Forensic Science International: Digital Investigation **48**, 301683 (2024). https://doi.org/https://doi.org/10.1016/j.fsidi.2023.301683, dFRWS EU 2024 - Selected Papers from the 11th Annual Digital Forensics Research Conference Europe
18. Michelet, G., Breitinger, F.: ChatGPT, Llama, can you write my report? An experiment on assisted digital forensics reports written using (local) large language models. Forensic Science International: Digital Investigation **48**, 301683 (Mar 2024). https://doi.org/10.1016/j.fsidi.2023.301683
19. Mohamed, H., Koroniotis, N., Schiliro, F., Moustafa, N.: IoT-CAD: A comprehensive Digital Forensics dataset for AI-based Cyberattack Attribution Detection methods in IoT environments. Ad Hoc Networks **174**, 103840 (Jul 2025). https://doi.org/10.1016/j.adhoc.2025.103840
20. Myneni, S., Jha, K., Sabur, A., Agrawal, G., Deng, Y., Chowdhary, A., Huang, D.: Unraveled — A semi-synthetic dataset for Advanced Persistent Threats. Computer Networks **227**, 109688 (May 2023). https://doi.org/10.1016/j.comnet.2023.109688
21. Nikolakopoulos, A., Evangelatos, S., Veroni, E., Chasapas, K., Gousetis, N., Apostolaras, A., Nikolopoulos, C.D., Korakis, T.: Large language models in modern forensic investigations: Harnessing the power of generative artificial intelligence in crime resolution and suspect identification. In: 2024 5th International Conference in Electronic Engineering, Information Technology & Education (EEITE). pp. 1–5 (2024). https://doi.org/10.1109/EEITE61750.2024.10654427
22. Ozkaya, I.: Application of large language models to software engineering tasks: Opportunities, risks, and implications. IEEE Software **40**(3), 4–8 (2023). https://doi.org/10.1109/MS.2023.3248401
23. Rajpurkar, P., Zhang, J., Lopyrev, K., Liang, P.: SQuAD: 100,000+ questions for machine comprehension of text. In: Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing. Association for Computational Linguistics, Stroudsburg, PA, USA (2016)
24. Scanlon, M., Breitinger, F., Hargreaves, C., Hilgert, J.N., Sheppard, J.: ChatGPT for digital forensic investigation: The good, the bad, and the unknown. Forensic Science International: Digital Investigation **46**, 301609 (Oct 2023). https://doi.org/10.1016/j.fsidi.2023.301609
25. Sharma, B., Ghawaly, J., McCleary, K., Webb, A.M., Baggili, I.: Forensicllm: A local large language model for digital forensics. Forensic Science International: Digital Investigation **52**, 301872 (2025). https://doi.org/https://doi.org/10.1016/j.fsidi.2025.301872, dFRWS EU 2025 - Selected Papers from the 12th Annual Digital Forensics Research Conference Europe
26. Sharma, B., Ghawaly, J., McCleary, K., Webb, A.M., Baggili, I.: ForensicLLM: A local large language model for digital forensics. Forensic Science International:

Digital Investigation **52**, 301872 (Mar 2025). https://doi.org/10.1016/j.fsidi.2025.301872
27. Shullani, D., Fontani, M., Iuliani, M., Shaya, O.A., Piva, A.: VISION: a video and image dataset for source identification. EURASIP Journal on Information Security **2017**(1), 15 (Oct 2017). https://doi.org/10.1186/s13635-017-0067-2
28. Sood, A.K., Zeadally, S., Hong, E.: The paradigm of hallucinations in ai-driven cybersecurity systems: Understanding taxonomy, classification outcomes, and mitigations. Computers and Electrical Engineering **124**, 110307 (2025). https://doi.org/https://doi.org/10.1016/j.compeleceng.2025.110307
29. Studiawan, H., Breitinger, F., Scanlon, M.: Towards a standardized methodology and dataset for evaluating LLM-based digital forensic timeline analysis (May 2025). https://doi.org/10.48550/arXiv.2505.03100
30. Tihanyi, N., Bisztray, T., Dubniczky, R.A., Toth, R., Borsos, B., Cherif, B., Jain, R., Muzsai, L., Ferrag, M.A., Marinelli, R., Cordeiro, L.C., Debbah, M., Mavroeidis, V., Jøsang, A.: Dynamic Intelligence Assessment: Benchmarking LLMs on the Road to AGI with a Focus on Model Confidence. In: 2024 IEEE International Conference on Big Data (BigData). pp. 3313–3321 (Dec 2024). https://doi.org/10.1109/BigData62323.2024.10825051, iSSN: 2573-2978
31. Tihanyi, N., Ferrag, M.A., Jain, R., Bisztray, T., Debbah, M.: CyberMetric: A Benchmark Dataset based on Retrieval-Augmented Generation for Evaluating LLMs in Cybersecurity Knowledge. In: 2024 IEEE International Conference on Cyber Security and Resilience (CSR). pp. 296–302 (Sep 2024). https://doi.org/10.1109/CSR61664.2024.10679494
32. Turing, A.M.: Computing machinery and intelligence (1950). In: Ideas That Created the Future, pp. 147–164. The MIT Press (Feb 2021)
33. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, L., Polosukhin, I.: Attention is all you need. In: Proceedings of the 31st International Conference on Neural Information Processing Systems. p. 6000–6010. NIPS'17, Curran Associates Inc., Red Hook, NY, USA (2017)
34. Wang, A., Singh, A., Michael, J., Hill, F., Levy, O., Bowman, S.: GLUE: A multi-task benchmark and analysis platform for natural language understanding. In: Linzen, T., Chrupała, G., Alishahi, A. (eds.) Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP. pp. 353–355. Association for Computational Linguistics, Brussels, Belgium (Nov 2018). https://doi.org/10.18653/v1/W18-5446
35. Wang, S., Long, Z., Fan, Z., Huang, X., Wei, Z.: Benchmark self-evolving: A multi-agent framework for dynamic LLM evaluation. In: Proceedings of the 31st International Conference on Computational Linguistics. pp. 3310–3328. Association for Computational Linguistics, Abu Dhabi, UAE (2025)
36. Wickramasekara, A., Densmore, A., Breitinger, F., Studiawan, H., Scanlon, M.: AutoDFBench: A Framework for AI Generated Digital Forensic Code and Tool Testing and Evaluation. In: Proceedings of the Digital Forensics Doctoral Symposium. pp. 1–7. ACM, Brno Czech Republic (Apr 2025). https://doi.org/10.1145/3712716.3712718
37. Wickramasekara, A., Scanlon, M.: A Framework for Integrated Digital Forensic Investigation Employing AutoGen AI Agents. In: 2024 12th International Symposium on Digital Forensics and Security (ISDFS). pp. 01–06. IEEE, San Antonio, TX, USA (Apr 2024). https://doi.org/10.1109/ISDFS60797.2024.10527235
38. Yin, Z., Wang, Z., Xu, W., Zhuang, J., Mozumder, P., Smith, A., Zhang, W.: Digital forensics in the age of large language models (2025), https://arxiv.org/abs/2504.02963