Forensic Event Reconstruction Using an Ontology-Based Approach and Large Language Model-Driven Reporting

Fitria Nurulhikmah[1]* and Hudan Studiawan[1]

[1]Department of Informatics, Institut Teknologi Sepuluh Nopember, Surabaya, Indonesia
6025232031@student.its.ac.id, hudan@its.ac.id
*Corresponding author

Abstract. Digital forensic investigations generate tens of thousands of artifacts originating from heterogeneous sources. This makes incident timeline reconstruction both time-consuming and error-prone. This study develops and evaluates an automated five-step pipeline for ontology-based digital incident timeline reconstruction. The pipeline is implemented on a public forensic dataset that includes a Windows 11 Enterprise disk image. The preprocessing stage classifies events from Plaso parsers. The ontology produces resource description framework (RDF) which spans three knowledge layers: the criminal knowledge Layer, the system knowledge layer, and the traceability knowledge layer. Ten forensic SPARQL queries are executed to extract key findings. In the final stage, six LLMs with three local models (tinyllama:1.1b, qwen2.5:0.5b, qwen2.5:1.5b) and three cloud-based models (llama-3.1-8b, qwen-2.5-7b, qwen-2.5-72b) are evaluated across three prompting scenarios using BLEU and ROUGE metrics. The cloud models achieve a mean BLEU/ROUGE score of 0.3088 compared to 0.1703 for their local counterparts (+81.3%), with cloud inference being approximately 8× faster. Few-shot prompting consistently outperforms both knowledge-augmented and baseline prompts across all cloud models. The findings offer empirical guidance for deploying LLMs in forensic workflows, including a quantifiable privacy trade-off between local and cloud inference.

Keywords: Digital Forensics · Timeline Reconstruction · Ontology · Large Language Models · BLEU/ROUGE · SPARQL

1 Introduction

The evolution of information technology has profoundly reshaped human life. While beneficial, it also creates new opportunities for cybersecurity incidents. The resulting volume and heterogeneity of digital artifacts, system logs, file metadata, browser histories, configuration files, pose significant challenges for accurately reconstructing forensic timelines [4].

© The Author(s) 2026
M. Murtadha et al. (eds.), *Proceedings of the 2026 11th International Conference on Sustainable Information Engineering and Technology (SIET 2026)*, Atlantis Highlights in Computer Sciences 26,
https://doi.org/10.2991/978-94-6239-828-3_20

Traditional forensic reconstruction is largely manual and prone to interpretive errors. On a single machine, hundreds to thousands of events per minute are scattered across operating systems, applications, and network services. Without structured methodologies and standardized data representation, the process becomes time-consuming, error-prone, and hard to replicate, challenges long recognized as problems of volume, heterogeneity, and legal reproducibility [1].

Ontology-based approaches, particularly the Ontology for the Representation of Digital Incidents and Investigations (ORD2I) [1], offer a pathway toward standardization. ORD2I organizes digital events chronologically and semantically through three layers: the criminal knowledge layer (CKL), system knowledge layer (SKL), and traceability knowledge layer (TKL), enabling automated reasoning and cross-artifact exploration via SPARQL.

Simultaneously, large language models (LLMs) have emerged as candidate tools for automating structured forensic reports. Michelet and Breitinger [2] demonstrated that LLMs can generate coherent reports from evidence summaries, but did not explore ontology-derived context or the critical question of local versus cloud deployment. Transmitting case evidence to external APIs risks violating chain-of-custody and legal confidentiality requirements.

The primary contribution is not a novel reconstruction algorithm but the systematic integration and first empirical quantification of ontology-based representation, knowledge-graph analysis, and LLM-driven reporting under an explicit local-versus-cloud comparison.

This study addresses existing gaps by formulating three research questions:

- RQ1: To what extent can an ontology-based workflow represent forensic timeline data?
- RQ2: How can SPARQL analysis be leveraged to reconstruct digital events?
- RQ3: What is an effective method for generating forensic investigation reports using Large Language Models?

Our contributions are fivefold: (1) an end-to-end automated pipeline for ontology-based incident timeline reconstruction on Windows 11; (2) an openly released RDF corpus of 539,475 triples from 31,878 multi-source events; (3) ten reproducible SPARQL forensic queries with documented limitations; (4) a systematic benchmark of six LLMs across three prompting scenarios evaluated with BLEU and ROUGE; and (5) an empirical analysis of the quality-privacy trade-off between local and cloud LLM deployment.

2 Related Work

ORD2I [1] provides three layers and analytical operators for automated timeline reconstruction. Seven efficiency criteria include volume handling, heterogeneity, and legal reproducibility. Its rich semantic representation and explicit operators make it suitable for legally-sound automation. A recent review of 29 forensic ontologies [5] offers recommendations for ontology development, particularly for reporting and tool integration. Earlier ontology-related work in security and

forensics includes an ontology for modeling computer attacks in intrusion detection [14] and the use of CybOX to standardize the representation and exchange of digital forensic information [15].

Michelet and Breitinger [2] were among the first to systematically apply LLMs to forensic report generation, and demonstrate that GPT-class models can produce structured reports from evidence summaries. Their work established the feasibility of automated reporting yet relied on larger proprietary cloud models without exploring ontology-derived context or comparing local alternatives. Studiawan et al. [3, 4] contributed a public forensic dataset and adopted BLEU and ROUGE metrics for quantitative quality evaluation, for reproducible comparisons. Our work builds on this evaluation framework while extending it in three directions: using ORD2I-derived forensic digests as structured LLM context, comparing six models spanning three orders of magnitude in parameter count, and introducing three prompting strategies to measure the effect of context design.

The trade-off between local and cloud LLMs has been examined in healthcare and legal domains where data sensitivity is critical [6]. In digital forensics, cloud APIs risk confidentiality and chain-of-custody; local models preserve privacy but their quality is unknown. This paper provides the first empirical quantification of this trade-off for forensic report generation.

No prior work integrates ORD2I-derived SPARQL digests as LLM context, benchmarks models across three parameter-count orders of magnitude, or quantifies the local-versus-cloud privacy-quality trade-off for forensic reporting, the gap this study addresses.

3 Methodology

3.1 Dataset and Experimental Setup

We employ Zenodo Scenario 1 [4], a publicly available Windows 11 Enterprise forensic image (26 December 2023, MIT license). The image was processed with log2timeline/Plaso [7] to produce `all_events.csv`. Experiments ran on Ubuntu 24.04 inside a VirtualBox VM (Intel Core i7-1065G7, 16 GB RAM; 4 cores and 8 GB allocated; no GPU). Local LLMs were served via Ollama [10]; cloud models via the OpenRouter API [13].[1] The evaluated models include the Qwen2.5 [11] and Llama 3.1 [12] families.

3.2 Workflow Overview

This study adopts a quantitative experimental approach implementing the ORD2I framework with one primary variable, LLM type (local versus cloud), and one controlled factor, three prompting scenarios (S1, S2, S3). Reports from each model-scenario combination are evaluated against a single ground-truth reference using BLEU [8], ROUGE-1, ROUGE-2, and ROUGE-L [9]. The pipeline

[1] https://openrouter.ai

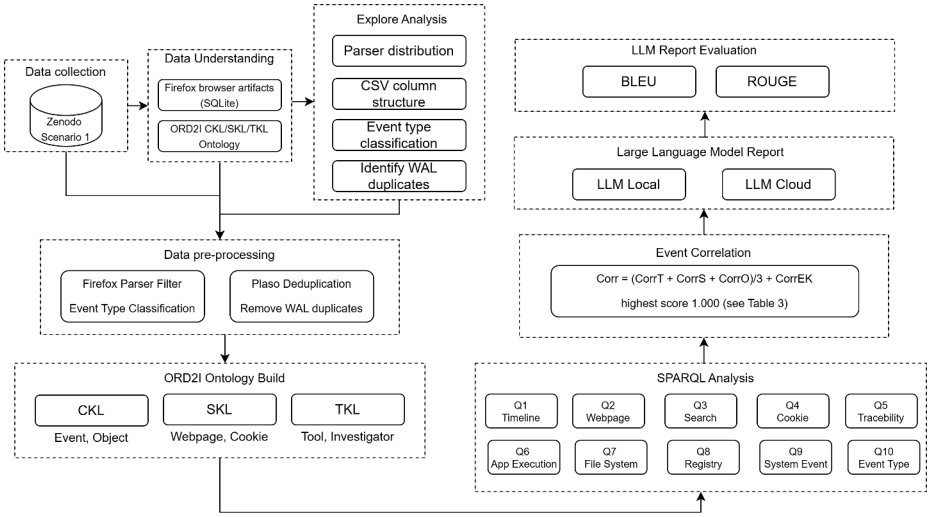


Figure 1. Five-step pipeline workflow

comprises five modular Python scripts executed sequentially (Figure 1 and Table 1). The complete codebase is publicly available at our GitHub repository: https://github.com/forensic-timeline/recononto.

3.3 Steps 1–2: Preprocessing and ORD2I Instantiation

Step 1 applies rule-based classification to assign one of 18 ORD2I-compatible event types to each Plaso event (e.g., *WebpageVisit, WebSearch, AppExecution, RegistryModify, FileCreate*). Additional fields (*url, host, page_title, app_name, registry_key, file_path*) are extracted from Plaso message fields to populate SKL entity properties.

Step 2 instantiates each event as an *ord2i:Event* using rdflib. The CKL encodes timestamps, event types, messages, parsers, and sources. The SKL instantiates referenced entities (26,944 *File*, 1,304 *RegistryKey*, 186 *Cookie*, 41 *Webpage*, 9 *WebResource*), each linked via *hasObject*. The TKL creates one *InvestigativeOperation* node (tool=Plaso, confidence=0.9), yielding 539,475 triples (≈16 per event).

Table 1. Five-Step Pipeline: Inputs, Processing, and Outputs

Step	Script	Input	Output	Key Result
1	01_preprocess_all.py	all_events.csv	31,878 classified events	18 event types, 26 parsers
2	02_build_ord2i_full.py	Classified events CSV	scenario1_ord2i_full.ttl	539,475 RDF triples (CKL, SKL,TKL)
3	03_sparql_analysis_full.py	TTL ontology file	CSV Q1–Q10 CSV result	10 forensic query outputs
4	04_event_correlation_full.py	Event CSV	correlation_results.csv	Top correlation: 1.000
5	05_llm_report_multimodel.py 05_llm_openrouter_tiers.py	Q1–Q10 and correlation	9 local + 9 cloud reports; unified eval CSV	6 models × 3 scenarios BLEU/ROUGE

3.4 Step 3: SPARQL Analysis

Ten SPARQL 1.1 queries (Q1–Q10) extract forensic dimensions: chronological timeline, webpage visits, search queries, cookie access, TKL traceability, application execution, file system operations, registry modifications, system events, and event type statistics. Three systematic limitations are documented: (i) WAL duplication in Plaso's SQLite processing causes 41 rows for 20 unique URLs in Q2; (ii) three misclassified URL rows appear in Q3 due to the *chrome_27_history* parser; (iii) Q9 *event_id* fields are empty due to a format mismatch between Plaso's *winevtx* output and ORD2I's *hasEventID* pattern. None of these affects the forensic reconstruction validity.

3.5 Step 4: ORD2I Event Correlation

Pairwise correlations are computed within temporal windows (300 s for browser events, 600 s for execution events) using [1]:

$$\text{Corr}(e_1, e_2) = (\text{CorrT} + \text{CorrS} + \text{CorrO})/3 + \text{CorrEK}$$

where $\text{CorrT} = e^{-|\Delta t|/W}$ measures temporal proximity via exponential decay; CorrS scores source similarity using a tiered category scheme (1.0 same category, 0.7 browser-execution, 0.6 execution-system, 0.4 browser-system, 0.3 otherwise); CorrO quantifies URL/path overlap via normalized longest common subsequence; and CorrEK applies expert rules (1.0 for *WebSearch* immediately preceding a matching non-search-engine *WebpageVisit*; 0.7 for *partial causal patterns*, i.e., pairs following the expected causal direction without a direct triggering relationship). Rule 1 includes a scenario-specific matching component (see Section 5.4).

3.6 Step 5: LLM Report Generation and Evaluation

A forensic digest of 2,197 characters was compiled from the outputs of Q1–Q10 and the top correlation pairs. The digest summarizes the incident into four labeled phases: Edge/Bing Activity (Phase 1), Firefox Installation (Phase 2), Firefox Browsing (Phase 3), and Session End (Phase 4). This digest was provided as context to each model under three prompting scenarios:

- S1 (Baseline): Digest plus a structured output template (executive summary / full timeline / key findings / conclusion).
- S2 (ORD2I-augmented): Digest plus top-5 correlation results (scores, Δt, CorrEK values, causal interpretations) with artifact source labels.
- S3 (Few-shot): Digest plus a complete annotated example report. The model was instructed to replicate the structure, not copy the example.

Ground-truth reports are prepared manually by researchers in consultation with research supervisors based on verified forensic artifacts and official scenario timelines, which serve as a consistent reference for all evaluations.

Table 2. Event Type Distribution: Top 10 of 18 Types

Event Type	Count	Percentage	Primary Parser(s)
FileCreate	11,167	35.0%	*usnjrnl*
FileSystemOp	9,833	30.8%	*usnjrnl / filestat*
FileModify	3,034	9.5%	*usnjrnl*
FileDelete	2,906	9.1%	*usnjrnl*
SystemEvent	1,908	6.0%	*winevtx*
RegistryModify	1,304	4.1%	*winreg/winreg_default*
FileDownload	573	1.8%	*chrome_cache*
CookieAccess	186	0.6%	*sqlite/chrome_66_cookies*; *firefox_10_cookies*
AppExecution	146	0.5%	*prefetch*
WebpageVisit	41	0.1%	*sqlite/chrome_27_history; firefox_history*
Other 8 types	780	2.4%	Various (*WebSearch, AppInstall, etc.)*
Total	**31,878**	**100%**	26 parsers, 7 sources

4 Experimental Results

4.1 Artifact Preprocessing and ORD2I Instantiation (Steps 1–2)

Step 1 classified 31,878 events into 18 types. File system operations dominate (*FileCreate* 35.0%, *FileSystemOp* 30.8%); Table 2 shows the top 10 types. Step 2 produced 539,475 RDF triples (approximately 16 per event), distributed across the CKL (31,878 *Event instances*), the SKL (26,944 *File entities*, 1,304 *RegistryKey* entries, 186 *Cookies,* 41 *Webpages*, and 9 *WebResources*), and one TKL *InvestigativeOperation* node. All ten SPARQL queries in Step 3 executed successfully and returned valid results.

4.2 SPARQL Queries and Forensic Findings (Step 3)

Analysis of Q1–Q10 reconstructed a 14 minute incident (00:34:22–00:48:28 UTC) comprise four distinct phases. From the query outputs, we identified eleven key forensic findings, which we label as F1 through F11 and group into three phases (Phases 1–3), while Phase 4 is described narratively. These findings are as follows:

- *F1–F3 (Phase 1 — Edge/Bing):* `MSEDGE.EXE` first executed at 00:34:24 (prefetch, run_count=1); Bing search for "mozilla firefox download" at 00:35:56; `Firefox Installer.exe` downloaded from `mozilla.org` at 00:37:07 (NTFS filestat).
- *F4–F7 (Phase 2 — Firefox Installation):* Installer executed at 00:39:14; `setup-stub.exe` recorded in Amcache at 00:41:49; 55+ files written to `C:\Program Files\Mozilla Firefox\` at 00:42:51–43:04; first `FIREFOX.EXE` execution at 00:44:03. Installation corroborated by three independent artifact types.
- *F8–F11 (Phase 3 — Firefox Browsing):* Privacy notice at 00:44:45; Google navigation at 00:45:24; search for "how to perform sql injection attack" at

Table 3. Top-4 ORD2I Correlations as Forensics Reconstruction Results

Score	Event Type 1	Event Type 2	Δt (s)	CorrEK	Forensic Significance
1.000	WebSearch	WebpageVisit	23	1.0	SQL injection search → W3Schools tutorial [causal chain]
0.866	WebpageVisit	WebpageVisit	0	0.0	HTTP→HTTPS redirect on google.com [navigation artifact]
0.844	WebpageVisit	WebSearch	21	0.7	Google opened → SQL injection query issued [browsing sequence]
0.823	WebSearch	CookieAccess	31	0.7	SQL injection search → W3Schools cookies set [independent corroboration]

Note: The perfect score (1.000) results from an expert rule for WebSearch → non-search-engine WebpageVisit; other trivial process-launch pairs also reach 1.000 without triggering CorrEK. The four pairs shown were selected for forensic relevance, not score alone.

00:45:45; W3Schools `sql_injection.asp` accessed at 00:46:08 ($\Delta t = 23$ s); `_ga` and `_sharedID` cookies set at 00:46:16.

– *Phase 4 — Session End:* `firefox.exe` last recorded by BAM registry at 00:48:14; UserLogon event at 00:48:26 indicates session termination.

Table 3 presents the four highest-scoring event pairs. The top correlation (score 1.000) identifies the causal chain between the SQL injection search and W3Schools access, driven by the expert rule CorrEK = 1.0. The 23-second interval and independent cookie evidence further support this finding.

4.3 LLM Evaluation Results (Step 5)

Table 4 reports BLEU and ROUGE scores for all 18 combinations, evaluated against a single unified ground-truth reference. Cloud models consistently and substantially outperformed local models: mean BLEU/ROUGE 0.3088 vs. 0.1703 (+81.3%). The best overall result was qwen-2.5-72b with S3 (Mean = 0.3607); among local models, qwen2.5:1.5b with S3 performed best (Mean = 0.2414). tinyllama (1.1B) failed on S2 and S3 (BLEU = 0.000), consistent with the baseline pattern observed throughout. Notably, qwen2.5:0.5b again scored best on S2 (0.2182) rather than S3 (0.1521), reinforcing that for models near the lower capability boundary, explicit ORD2I context may substitute for the structural guidance larger models derive from few-shot examples. Cloud inference averaged ≈14s per report versus ≈112s for local models (≈8× faster).

Table 4. LLM Evaluation Results (BLEU/ROUGE, Unified Ground Truth)

Model	Scenario	BLEU	ROUGE-1	ROUGE-2	ROUGE-L	Mean	Time (s)
tinyllama (1.1B)	S1	0.0451	0.3186	0.1797	0.2219	0.1913	140.9
	S2	0.0000	0.1272	0.0223	0.0604	0.0525	74.6
	S3	0.0000	0.2183	0.0204	0.0964	0.0838	131.7
qwen2.5 (0.5B)	S1	0.0657	0.3892	0.1463	0.2177	0.2047	73.6
	S2	0.0764	0.4433	0.1591	0.1940	0.2182	89.8
	S3	0.0306	0.3385	0.0854	0.1538	0.1521	115.0
qwen2.5 (1.5B)	S1	0.0412	0.4046	0.1089	0.1896	0.1861	131.2
	S2	0.0703	0.4207	0.1343	0.1857	0.2027	125.2
	S3	0.0980	0.4521	0.1650	0.2506	**0.2414**	124.0
llama-3.1 (8B)	S1	0.2059	0.4472	0.2414	0.3119	0.3016	9.3
	S2	0.1423	0.4640	0.2257	0.2568	0.2722	7.4
	S3	0.2164	0.4506	0.2701	0.3113	0.3121	7.2
qwen-7b (7B)	S1	0.2022	0.4661	0.2622	0.3020	0.3081	17.6
	S2	0.1640	0.4229	0.2251	0.2676	0.2699	12.7
	S3	0.1959	0.4562	0.2816	0.3145	0.3120	21.6
qwen-72b (72B)	S1	0.1848	0.5310	0.3165	0.3509	0.3458	17.8
	S2	0.1505	0.4781	0.2681	0.2911	0.2970	15.2
	S3	0.2515	0.5065	0.3179	0.3669	**0.3607**	16.9
Local average				0.1703			111.78
Cloud average				0.3088			13.97

Scenarios: S1 = no additional knowledge, S2 = with ORD2I context, S3 = few-shot template. BLEU/ROUGE is calculated against a single ground-truth report applied equally to all models (Section 3.6). No confidence intervals are reported since each combination was evaluated on a single test instance; results are indicative rather than statistically rigorous.

5 Discussion

5.1 RQ1–2: Effectiveness of the ORD2I Workflow

The workflow answers RQ1 affirmatively: fully automated multi-source timeline reconstruction is achievable with ORD2I on Windows 11 Enterprise at scale. The 539,475-triple knowledge graph enables unified SPARQL querying across seven artifact types, exposing relationships invisible to single-artifact tools. The causal chain score of 1.000 (SQL injection search → W3Schools, CorrEK = 1.0) shows the correlation engine can identify forensically meaningful relationships, though broader precision/recall evaluation is needed. The three-source corroboration of Firefox installation satisfies the forensic corroboration principle, answering RQ2. The three documented limitations (WAL duplication, URL misclassification, empty event_id) are systematic, reproducible, and suitable to note in forensic reports.

5.2 RQ3: Capability of Local LLMs

The results confirm a clear capability threshold around 1.5B parameters for this task. tinyllama (1.1B) cannot reliably follow complex multi-phase prompts: it fails completely on S2 and S3 (BLEU = 0.000) and echoes the digest on S1 rather than synthesizing a report. qwen2.5:1.5b represents the minimum viable local model (Mean 0.2414, S3), and produce structured reports covering seven of

the ten forensic phases across all three scenarios. Notably, qwen2.5:0.5b achieves its best score on S2 (0.2182) rather than S3, suggesting that for models at the lower capability boundary, explicit ORD2I correlation context substitutes for the structural guidance that larger models derive from few-shot examples. In direct answer to RQ3, effectiveness depends on model capacity: few-shot prompting (S3) suits models $\geq$1.5B, while explicit ORD2I context (S2) suits smaller models.

5.3 Privacy Trade-off and Practical Implications

Transmitting forensic artifacts to external cloud APIs introduces legal and chain-of-custody risks that may be unacceptable in active criminal investigations or under strict data protection regimes (e.g., GDPR). The public-domain dataset used here incurs no such risk, but practitioners must exercise caution with real case evidence. Our results support a two-tier decision framework: (i) for privacy-sensitive investigations, local qwen2.5:1.5b (1 GB, no network dependency, Mean 0.2414 on S3) is the best available privacy-preserving option, albeit with a wider quality gap than previously estimated; (ii) for research or pre-anonymized evidence, cloud qwen-2.5-72b on S3 (Mean 0.3607) delivers the highest quality with fast inference ($\sim$17s). This trade-off ($+$81.3% mean BLEU and ROUGE favoring cloud) is larger than initially estimated, so practitioners choosing local deployment for privacy should anticipate a more pronounced quality reduction than a smaller gap would imply.

5.4 Limitations

This study has several limitations. The pipeline was evaluated on a single Zenodo scenario using one Windows 11 dataset and a single manually prepared ground-truth report; therefore, its generalizability to other operating systems, forensic scenarios, and user activities remains untested. BLEU and ROUGE were adopted for consistency with prior work but remain susceptible to lexical reference bias. As noted in Table 4, single-instance evaluation precludes formal statistical significance testing, and no external forensic expert validated the generated reports. Finally, the CorrEK expert rules, particularly Rule 1, include scenario-specific vocabulary, which may limit applicability to unseen forensic scenarios.

6 Conclusion and Future Work

We presented an automated five-step pipeline for digital incident timeline reconstruction using the ORD2I ontology on Windows 11 Enterprise artifacts. Applied to the public Zenodo Scenario 1 dataset, the pipeline processed 31,878 multi-source events into 539,475 RDF triples, executed ten SPARQL queries identifying eleven key findings, and computed event correlations yielding a perfect-score causal chain (1.000) linking an SQL injection search to a tutorial page access with three independent corroborating sources. The 14-minute session reflects

deliberate reconnaissance: a fresh Firefox installation immediately followed by targeted SQL injection research.

LLM evaluation across six models and three scenarios showed cloud-based qwen-2.5-72b (S3) achieved the highest mean BLEU/ROUGE score (0.3607), while local qwen2.5:1.5b (S3, 0.2414) was the best privacy-preserving alternative, with cloud inference ≈8× faster. These results offer actionable guidance for practitioners choosing between local and cloud LLM deployment. Future work will extend the evaluation to all publicly available Zenodo scenarios, incorporate semantic similarity metrics (e.g., SBERT), and include validation by human experts.

Acknowledgement. This research is funded by Institut Teknologi Sepuluh Nopember (ITS) under the Strategic Research Grant (SRG) Type D Scheme (Contract No. 1666/PKS/ITS/2026). The authors thank the researchers who made the Zenodo Scenario dataset publicly available, as well as the Plaso, Ollama, and OpenRouter development teams.

References

1. Chabot, Y., Bertaux, A., Nicolle, C., Kechadi, T.M.: An ontology-based approach for the reconstruction and analysis of digital incidents timelines. Digit. Investig. **15**, 83–95 (2015). https://doi.org/10.1016/j.diin.2015.07.001
2. Michelet, L., Breitinger, F.: Can large language models write forensic reports? In: Proceedings of DFRWS EU (2024)
3. Studiawan, H., Breitinger, F., Scanlon, M.: Towards a standardized methodology and dataset for evaluating LLM-based digital forensic timeline analysis. Forensic Science International: Digital Investigation **54**, 301982 (2025). https://doi.org/10.1016/j.fsidi.2025.301982
4. Studiawan, H., Sohel, F., Payne, C.: Automated digital forensics report generation using large language models: dataset and evaluation. Zenodo (2025). https://doi.org/10.5281/zenodo.15493424
5. Silva, T., Oliveira Jr., E., Zorzo, A.: How ontologies have supported digital forensics: review and recommendations. Forensic Science Review, 99–125 (2024)
6. Tang, R., Han, X., Jiang, X., Hu, X.: Does synthetic data generation of LLMs help clinical text mining? arXiv preprint arXiv:2303.04360 (2023)
7. Garfinkel, S., Malan, G., Schatz, B.: log2timeline/plaso — super timeline all the things. https://github.com/log2timeline/plaso (2024)
8. Papineni, K., Roukos, S., Ward, T., Zhu, W.: BLEU: a method for automatic evaluation of machine translation. In: Proceedings of ACL, pp. 311–318 (2002)
9. Lin, C.Y.: ROUGE: a package for automatic evaluation of summaries. In: Text Summarization Branches Out Workshop, ACL 2004, pp. 74–81 (2004)
10. Ollama: Run large language models locally on your machine. https://ollama.com (2024)
11. Qwen Team: Qwen2.5 technical report. arXiv preprint arXiv:2412.15115 (2024)
12. Meta AI Research: The Llama 3 herd of models. arXiv preprint arXiv:2407.21783 (2024)
13. OpenRouter: Unified API interface for large language models. https://openrouter.ai (2024)
14. Undercoffer, J., Joshi, A., Pinkston, J.: Modeling computer attacks: an ontology for intrusion detection. In: RAID 2003, pp. 113–135 (2003)
15. Casey, E., Back, G., Barnum, S.: Leveraging CybOX™ to standardize representation and exchange of digital forensic information. Digit. Investig. **12**, S102–S110 (2015)

Open Access This chapter is licensed under the terms of the Creative Commons Attribution-NonCommercial 4.0 International License (http://creativecommons.org/licenses/by-nc/4.0/), which permits any noncommercial use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license and indicate if changes were made.

The images or other third party material in this chapter are included in the chapter's Creative Commons license, unless indicated otherwise in a credit line to the material. If material is not included in the chapter's Creative Commons license and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder.

