ForensicsData: a Digital Forensics Dataset for Large Language Models

Youssef Chakir*, Iyad Lahsen-Cherif*
*INPT, CS department, Rabat, Morocco.
Emails: chakir.youssef@master.inpt.ac.ma, lahsencherif@inpt.ac.ma

***Abstract*—The growing complexity of cyber incidents presents significant challenges for digital forensic investigators, especially in evidence collection and analysis. Public resources are still limited because of ethical, legal, and privacy concerns, even though realistic datasets are necessary to support research and tool developments. To address this gap, we introduce ForensicsData, an extensive Question-Context-Answer (Q-C-A) dataset sourced from actual malware analysis reports. It consists of more than 5,000 Q-C-A triplets. A unique workflow was used to create the dataset, which extracts structured data, uses large language models (LLMs) to transform it into Q-C-A format, and then uses a specialized evaluation process to confirm its quality. Among the models evaluated, Gemini 2 Flash demonstrated the best performance in aligning generated content with forensic terminology. ForensicsData aims to advance digital forensics by enabling reproducible experiments and fostering collaboration within the research community.**

Keywords: Digital Forensics, Malware Analysis, Synthetic Data Generation, Large Language Models, Question-Context-Answer (Q-C-A) Datasets.

I. Introduction

Digital forensics has emerged as a critical discipline in modern cybersecurity, focusing on the systematic collection, preservation, examination, and analysis of digital evidence to support legal proceedings and incident response. As digital devices become ubiquitous, forensic investigators face mounting challenges in managing vast quantities of heterogeneous data, making traditional manual analysis methods labor-intensive and error-prone. The standardized forensic process involves identifying evidence sources, preserving data integrity through forensic imaging, collecting information with specialized tools, conducting detailed examinations, and producing legally admissible reports. However, the development and validation of digital forensic tools face a significant bottleneck: the scarcity of realistic, publicly available datasets for training and testing purposes. This limitation stems from stringent privacy regulations, legal restrictions on data sharing, and the inherently sensitive nature of forensic evidence. Consequently, researchers struggle to access sufficient training data, hampering the development of robust forensic tools and limiting research reproducibility. This challenge is particularly acute in malware analysis, where dynamic threats and evolving attack techniques demand continuous updates to detection capabilities.

The emergence of Large Language Models (LLMs) represents a paradigm shift in artificial intelligence, offering unprecedented capabilities in natural language understanding and generation. Built on transformer architectures and trained on extensive text corpora, state-of-the-art models such as GPT-4, Claude, LLaMA, and Gemini demonstrate remarkable proficiency in complex tasks including text synthesis, summarization, anomaly detection, and information retrieval. In digital forensics, LLMs show promising applications in automating evidence triage, streamlining malware analysis workflows, generating comprehensive investigative reports, and detecting anomalous patterns in digital artifacts.

A particularly compelling advantage of LLMs lies in their ability to generate realistic synthetic datasets that preserve the linguistic and structural properties of authentic forensic data. This capability addresses the critical challenge of dataset scarcity by enabling the creation of training and testing resources without relying on sensitive real-world evidence. Synthetic datasets can maintain the complexity and diversity necessary for robust tool development while circumventing ethical, legal, and privacy constraints. Despite these promising developments, fundamental questions remain unanswered regarding the effective application of LLMs in digital forensics dataset generation. The quality, diversity, and accuracy of synthetic datasets must be rigorously evaluated to ensure their utility for training forensic tools. This research addresses these challenges through a comprehensive investigation of LLM-based synthetic dataset generation for digital forensics, with particular emphasis on malware behavior analysis. Specifically, we evaluate and validate the quality, realism, and effectiveness of our newly introduced **ForensicsData** dataset. We focus on three primary research questions:

1) **Dataset Quality and Realism**: Can LLMs generate realistic, diverse, and accurate synthetic malware behavior datasets that effectively capture the complexity and variability of real-world threats?
2) **Comparative Model Performance**: How do different LLM architectures compare in terms of generation quality, accuracy, efficiency, and cost-effectiveness for forensic dataset creation?
3) **Validation Methodologies**: What validation techniques are most effective for ensuring the quality, reliability, and forensic relevance of synthetic datasets?

To address these questions, we present **ForensicsData**, a comprehensive digital forensics Question-Context-Answer (Q-C-A) dataset derived from contemporary malware analysis reports. This dataset represents the first publicly available, structured Q-C-A resource specifically designed for digital forensics applications, comprising over 5,000 annotated triplets extracted from malware reports published in 2025. Each entry encompasses critical forensic information including malware metadata, behavioral patterns, indicators of compromise (IOCs), tactics, techniques, and procedures (TTPs), and mitigation strategies.

Our research makes three significant contributions to the digital forensics research community:

1) **Novel Dataset Creation**: We introduce **Forensics-**

979-8-3503-9281-4/25/$31.00 ©2025 IEEE

Data, a comprehensive synthetic dataset that accurately reflects malware behavior patterns across diverse Windows-based execution environments. The dataset incorporates rich Question-Context-Answer annotations specifically tailored for digital forensics applications, providing a valuable resource for training and evaluating forensic analysis tools.

2) **Scalable LLM-Driven Annotation Pipeline**: We propose and implement an innovative, scalable pipeline that leverages multiple state-of-the-art LLMs to semantically annotate malware reports with structured forensic insights. This pipeline incorporates advanced prompt engineering techniques, parallel processing capabilities, and robust error handling mechanisms to ensure consistent, high-quality output generation across diverse malware families and attack vectors.
3) **Comprehensive Validation Framework**: We develop and apply a multi-layered validation methodology that combines automated quality assessment techniques with expert evaluation protocols. This framework encompasses format validation, semantic deduplication, similarity filtering, and LLM-as-Judge evaluation to ensure the reliability, accuracy, and forensic relevance of generated datasets

The remainder of this paper is organized as follows: Section II reviews related work in digital forensics, synthetic data generation, and LLM applications in cybersecurity. Section III details our data collection methodology and preprocessing procedures. Section IV presents our comprehensive methodology for dataset generation and validation. Section V discusses experimental results and performance comparisons across different LLM architectures, findings, limitations, and implications. Section VI concludes with a summary of contributions and outlines future research directions.

II. Related Work

Digital forensics (DF) refers to the collection, examination, and presentation of digital evidence. Technically and legally, the digital forensic landscape is becoming more complex due to the rapid proliferation of technologies like cloud services, embedded systems, and the Internet of Things (IoT) [1]–[3]. Traditional forensic techniques, which were first created for more homogeneous and less voluminous data sources, are put to the test by the massive volumes of heterogeneous data generated by these contemporary computing environments [4], [5].

The lack of datasets appropriate for training and research is a major problem in digital forensics. Due to ethical, privacy, and legal restrictions, researchers are forced to use synthetic datasets instead of authentic forensic data. A good substitute that preserves privacy and permits scalable and repeatable experiments is the use of synthetic datasets [4], [6]–[9]. GPT and LLaMA, two recent developments in Large Language Models (LLMs), have shown promise in producing realistic and contextually rich synthetic forensic data [10], [11]. The practical implementation of these generative capabilities in digital forensic workflows is demonstrated by models such as *ForensicLLM* [6] and methods for producing comprehensive forensic reports [12]. Furthermore, the synthesis of logs simulating cyber threats [13] and the integration of LLM outputs with anomaly detection frameworks and explainable AI tools [1], [14] demonstrate promising avenues for automating threat triage, investigation, and analysis.

It is crucial to guarantee the stability and dependability of synthetic forensic data produced by LLMs. This calls for exacting standards and evaluation procedures. Initiatives like SciFaultyQA [15], CyberMetric [16], and CTIBench [17] offer structured datasets for assessing the fault detection, cybersecurity domain knowledge, and threat intelligence reasoning skills of LLM-generated outputs. Tools like LongCite [18] and LongWriter [10], which are expressly made to improve coherence in long-context text, are essential for enhancing the quality and accuracy of forensic reports that are produced, especially when it comes to citation accuracy and traceability. Maintaining the evidential rigor and admissibility necessary for forensic documentation requires the use of citation-aware frameworks, as demonstrated by the work of Gao et al. [19].

Additionally, cross-domain developments offer important insights on enhancing LLM outputs in digital forensics. Advances in fields like AI safety [20], recommender systems [21], and code synthesis [22] provide transferable approaches to guarantee safety, explainability, and controllability in LLM-driven synthetic data generation. Furthermore, domain-specific data augmentation methods investigated in intricate domains like chip design [23] demonstrate how LLMs may be tailored to certain technical contexts, indicating that DF may have a comparable potential.

Despite their considerable promise, LLMs and synthetic datasets face persistent challenges, including factual consistency, rigorous evaluation, and legal admissibility in digital forensics. Addressing these challenges requires sustained interdisciplinary research to align AI-driven tools with forensic standards, ensuring both technological innovation and adherence to legal frameworks.

While existing datasets have significantly contributed to digital forensic research, none specifically address malware behaviors in structured Question-Context-Answer (Q-C-A) format as comprehensively as **ForensicsData**. This gap underscores the importance and novelty of our contribution in facilitating structured forensic investigations and model training.

III. Data Collection

To create a comprehensive and up-to-date database of malware behavior, we sourced 1,500 execution reports from the publicly accessible *ANY.RUN*[1] malware analysis platform. It provides interactive sandbox environments for dynamic malware analysis and hosts over 10 million user-contributed malware and benign execution traces. The platform offers detailed logs of process execution, file system activity, network communications, and behavioral indicators associated with both malicious and benign software.

A. Data Source

The reports selected for this study were constrained to samples submitted in 2025 to ensure the relevance and currency of behavioral patterns. The dataset includes:

- **15 Malware Families**: covering a diverse range of threats (e.g., Remote Access Trojans, credential stealers, ransomware, and loaders).

[1]https://any.run/

- **Benign Samples**: merged to support comparative analysis and multi-label classification tasks.
- **Uniform Distribution**: across families to minimize class imbalance and improve the generalizability in the dataset.

TABLE I
MALWARE CLASSES

Malware Type	Malware Family
Banker	qbot trickbot
Benign	benign
Botnet	sality
Infostealer	formbook hawkeye
Infostealer/Loader	amadey
Ransomware	gandcrab wannacry
RAT	nanocore xworm remcos
Stealer	agenttesla lumma
Trojan	emotet

TABLE II
DISTRIBUTION OF MALWARE FAMILY

Malware Type	File Count	Percentage
agenttesla	73	6.6%
amadey	40	3.6%
emotet	33	3.0%
gandcrab	100	9.1%
lumma	47	4.3%
nanocore	76	6.9%
sality	83	7.5%
trickbot	64	5.8%
formbook	86	7.8%
hawkeye	54	4.9%
qbot	69	6.3%
remcos	69	6.3%
wannacry	36	3.3%
xworm	72	6.5%
benign	150	13.6%

B. Data Extraction and Preprocessing

The original reports were available in HTML and semi-structured XML formats. To prepare them for LLM processing, we performed the following preprocessing steps:

- **Parsing and Extraction:** Leveraging a custom pipeline built using *BeautifulSoup* and *lxml*, we extracted relevant behavioral sections from each report, including metadata (hashes, file names, verdicts, tags), indicators of compromise, behavior activities, behavior graphs, network indicators, and process trees.
- **Noise Removal:** Background processes unrelated to malware behavior were identified and removed. Files without meaningful content or containing incomplete information were also filtered out to reduce ambiguity.
- **Standardization**: All reports were converted to a consistent `JSON format` with standardized filed names and value formats to facilitate automated processing.
- **De-duplication**: The dataset was checked for repeated samples, and removed duplication malware report based on SHA-256 hash comparisons.

The preprocessing phase resulted in clean, structured dataset in JSON format (see Figure 1) ready for Question and Answer generation by LLMs

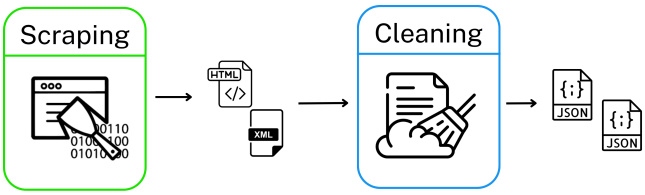

Fig. 1. Data Scraping and Cleaning

IV. METHODOLOGY

The core objective of this work is to transform raw malware execution reports into a high-quality synthetic dataset of question-context-answer (Q-C-A) triples for digital forensics research. To achieve this, we designed a multi-phase methodology comprising structured data transformation, semantic annotation via large language models (LLMs), and rigorous validation of outputs. The methodology is organized into two main components:

A. The LLM Annotation Pipeline and Validation Techniques.

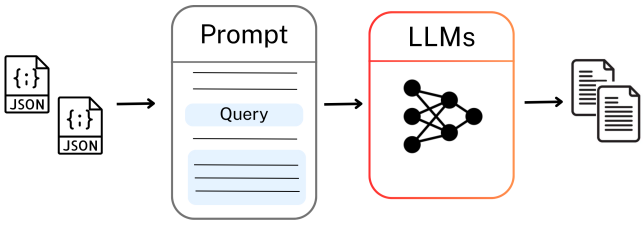

Fig. 3. Create Question Context Answer by a LLM

1) LLMs Pipeline: : The **ForensicsData** generation process involves transforming structured JSON representations of malware reports into Question-Context-Answer (Q-C-A) triples through a scalable and parallelized LLM annotation pipeline. The pipeline consists of the following stages (see Figure 3):

2) Prompt Engineering: : Each JSON report was passed to an LLM along with a prompt instructing it to generate contextually grounded Question-Context-Answer (Q-C-A) triples. Prompts were designed with care to return diverse forensic insights essential for **ForensicsData** across five dimensions.

- Malware identification and metadata.
- Technical indicators of compromise.
- Suspicious Behavioral patterns and techniques.
- Malicious Behavioral patterns and techniques.
- Impact assessment and mitigation strategies.

3) LLM Selection and Allocation: : To capture diversity in generation style and reasoning patterns, five different LLMs were used:

TABLE III
COMPARISON OF LANGUAGE MODELS

Model	Parameters	Context
Mistral 8B	≈ 7.3 billion	32k tokens
LLaMA 3–70B	≈ 70 billion	8k tokens
DeepSeek V3	33 billion	128k tokens
Qwen-QWQ-32B	≈ 32 billion	32k tokens
Gemini 2.0 Flash	Not disclosed	1M tokens

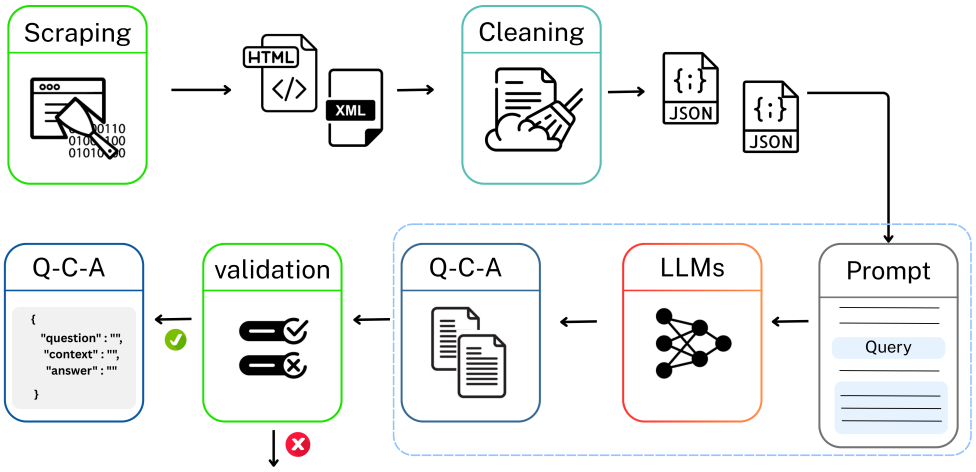

Fig. 2. pipeline for generating structured datasets Question-Context-Answer (Q-C-A) triples

Each model processed 20% of the dataset (approximately 300 reports) and generated five Q-C-A triples per report. This stratified approach ensured balanced representation across models and malware families.

Output Processing: The generated Q-C-A triples were automatically aggregated and formatted according to the predefined JSON schema, resulting in approximately 5,000 initial Q-C-A triples (1000-1200 Q-C-A triples per model). These triples were compiled into the **ForensicsData** dataset. The pipeline incorporated error handling and retry mechanisms to address generation failures or unexpected output

B. Multi-layered Validation for Synthetic Datasets

Given the synthetic nature of the dataset, validating the quality, coherence, and utility of generated Q-C-A triples was essential. We employed a multi-layered validation strategy:

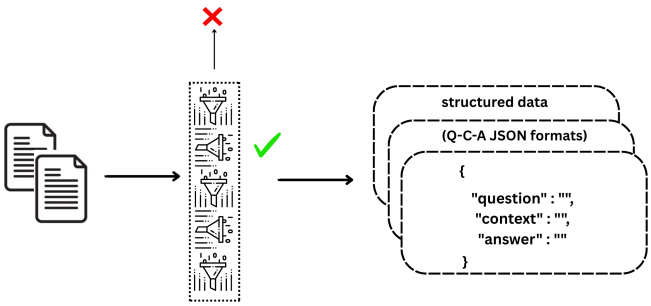

Fig. 4. Framework of Validation Dataset Generation

1. Format Validation via Pydantic library

Each generated Q-C-A entry was parsed using the Pydantic library to enforce schema correctness, type safety, and structural consistency across the dataset. The schema enforced the following structure:

```
{
  "question": "string",
  "context": "string",
  "answer": "string"
}
```

2. Deduplication and Similarity Filtering

We employed vector-based similarity analysis using the `all-MiniLM-L6-v2` Embedding Model from the `sentence-transformers` library to generate dense sentence embeddings. Cosine similarity scores, computed via scikit-learn, were used to identify questions with high semantic overlap. Questions with a similarity score above 0.9 were flagged and removed to eliminate redundancy. Given a Question Q, the `all-MiniLM-L6-v2` model produces an embedding vector $\mathbf{e}_q \in \mathbb{R}^d$, where $d = 385$ is the embedding dimension.

Cosine similarity between two embedding vectors $\mathbf{e}_a$ and $\mathbf{e}_b$ is computed as:

$$\text{cosine\_similarity}(\mathbf{e}_a, \mathbf{e}_b) = \frac{\mathbf{e}_a \cdot \mathbf{e}_b}{\|\mathbf{e}_a\| \|\mathbf{e}_b\|} \tag{1}$$

where, $\cdot$ denotes the dot product (inner product) of the two vectors, and $\|\mathbf{e}_a\|$ and $\|\mathbf{e}_b\|$ are their Euclidean norms.

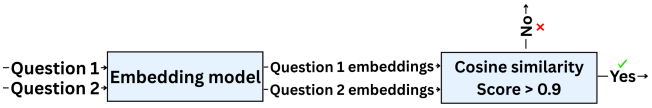

Fig. 5. Workflow for Semantic Questions Filtering Using Embedding Model

3. LLM-as-Judge Evaluation A high-performing language model (**Gemini Advanced 2.0 flash**) was used as an automated evaluator to assess the quality of question-context-answer triples. The evaluation followed a structured prompt that guided the LLM to analyze each Q-C-A triples across four detailed criteria:

- **Logical Validity and Realism** Determines whether the question is logically sound and based on realistic technical or security scenarios.
- **Relevance** Assesses if the answer directly addresses the question in a meaningful and appropriate way.
- **Completeness** Evaluates whether the answer fully covers all aspects of the question.
- **Consistency and Absence of Hallucinations** Verifies that the answer is internally consistent and free from hallucinated (fabricated) information.

For each criterion, the model provided a clear verdict (Yes/No, 1/0) along with a brief justification. Q-C-A triples that failed to meet one or more of the criteria were either revised or excluded from the dataset to maintain quality and reliability.

V. Results and Discussion

Here, we compare the performance of the five large language models used to produce **ForensicsData** dataset from formal JSON reports. Each model was responsible for enriching approximately 20% of the dataset semantically by producing five Q-C-A triples per report. We assess the usability of this generation process in the real world.

A. *Performance Benchmarking of LLMs*

To establish the pipeline operation efficiency of the pipeline in terms of generating datasets, we compared all the five LLMs on four aspects:

1) Latency: Comparing with models on host traditional infrastructures, GroqCloud-supported larger models such as the Qwen QWQ 32B and LLaMA 3-70B exhibit considerably lower latency and higher throughput. GroqCloud's use of application-specific hardware (GroqChips) designed for ultra-low latency inference is the performance advantage here. In contrast, vendors such as Mistral and Gemini rely on NVIDIA GPUs (such as the A100) and TPUs (such as the TPUv4), respectively, which are not as latency-optimized as Groq's own hardware.

2) Model Throuphput Performance: Qwen QWQ-32B and LLama3 70B achieved the highest throughput, supported by the optimized inferenced infrastructure provided by GroqCLoud.

3) Models Cost: Maximum output at cost-effective price is provided by the Qwen-QWQ-32B with the best trade-off between API cost and performance of output. The most cost-efficient models, Mistral 8B and Gemini Flash, are suited for usage scenarios where money is tight. In contrast, though LLaMA 3–70B excels in performance, it has the highest cost of output and is therefore not ideally suited for applications where cost-effectiveness takes priority.

4) Output format validity: Over 97% of generated outputs from every model were successfully validated by Pydantic schema, illustrating the success of structural formatting and prompt engineering. Compliance with instructions, however, was inconsistent: while Mistral 8B produced syntactically correct output, it occasionally failed to cover all semantic meaning of prompts, showing weaker instruction-following skills in spite of its structural correctness.

B. *Synthetic Dataset Quality Control and Validation*

In order to assure the usability and integrity of the resulting synthetic **ForensicsData** dataset produced by the five large language models, we performed a series of quality control procedures ensuring format validity, semantic variety, logical consistency, and redundancy. The following is a summary of the main dataset characteristics resulting from these post-generation tests.

1) Format Validation via Pydantic library: 98.68% of the entries passed schema validation without requiring any modification. The remaining entries were either auto-corrected or discarded if invalid.

2) Deduplication and Similarity Filtering: Deduplication and similarity filtering with the MiniLM-L6-V2 model at 0.90 cosine similarity threshold ensured that no Q-C-A triples were marked as duplicates. This indicates that the LLMs generated semantically different, context-sensitively aware content consistently even in similar malware samples, indicating high semantic diversity and ability to generate different outputs even in highly similar input scenarios.

Criterion	Q-C-A Passing	Validation Rate
Logical Validity	5,000 / 5,000	100.0%
Relevance	5,000 / 5,000	100.0%
Completeness	4850 / 5,000	98.0%
Hallucination-Free	5,000 / 5,000	100.0%

TABLE V
Validation results of Q-C-A generation

3) Logical Coherence Check via LLM-as-Judge: These findings confirm that the vast majority of Q-C-A triples are logically correct, accurate, and contextually applicable. The sole meaningful constraint was completeness—around 2% of answers left questions partially answered, typically glossing over technical details like file paths, process IDs, or behavior descriptions. Gemini 2.0 Flash, a validator based on an LLM, verified 100% logical correctness and hallucination-free generation and also detected that 2% of triples were completely complete. This shortfall offers potential for improvement through more directed prompting or post-generation revision.

C. *Strengths and Practical Implications*

The methodology and resulting dataset present several notable strengths:

1) **Realistic Behavioral Coverage**: Reports from 15 malware families ensure coverage across multiple malware categories, including RATs, Banker, Trajon, stealers, botnet, and ransomware.
2) **Recency and Relevance**: All source reports were collected in 2025, aligning the dataset with modern malware techniques and infrastructure (e.g., C2 communication patterns, evasion tactics).
3) **Structured and Validated Outputs**: JSON-based Q-C-A triples were schema-validated using Pydantic, and semantic coherence was verified using an independent LLM judge (Gemini 2.0 flash).
4) **Cross-Model Generation**: Using five LLMs improved stylistic and conceptual variety, enhancing the dataset's richness and generalizability.
5) **Ethical Safety**: By generating descriptive Q-C-A triples rather than executable code or raw malware, the pipeline maintains ethical integrity while enabling security research.

D. *Limitations and Challenges*

Despite its strengths, the approach has certain limitations:

1) **Instruction Sensitivity**: Some models, particularly Mistral 8B, adhered poorly to detailed instructions, requiring additional filtering or re-generation in post-processing.
2) **Source Dependency**: Output quality is inherently limited by the completeness and clarity of the original ANY.RUN report content.

E. *Hallucination and Bias Considerations*

LLMs are known to introduce hallucinations in generated outputs. We addressed these issues explicitly.

Hallucination Mitigation: While Gemini 2.0 Flash found no hallucinations in Q-C-A triples, we proactively implemented the following safeguards:

- **Prompt constraints**: LLMs were instructed to only use content available in the report JSON.

TABLE IV
PERFORMANCE BENCHMARKING OF LLMS USED FOR GENERATING **FORENSICSDATA**

Model	Provider	Parameters	Context Windows	Latency (s)	Cost Input/Output ($/1K tokens)	JSON Validity (%)
Qwen-QWQ-32B	GroqCloud	32B	32k	250	0.00029 / 0.00039	99.3
LLaMA 3–70B	GroqCloud	70B	8k	364	0.00059 / 0.00079	98.6
Gemini 2.0 Flash	Google	Undisclosed	1M	425	0.0001 / 0.0004	99.8
Mistral 8B	OpenRouter	7.3B	32k	750	0.0001 / 0.0001	97.2
DeepSeek V3	OpenRouter	33B	128k	1667	0.00038 / 0.00089	99.5

- **Schema enforcement**: Ensured structured and bounded outputs.
- **LLM-as-Judge filtering**: Discarded or revised outputs judged to contain inconsistencies.

VI. CONCLUSION AND PERSPECTIVES

The article describes a scalable approach to creating a synthetic malware behavior **ForensicsData** dataset for digital forensics that addresses dataset scarcity while maintaining privacy compliance. Using 1,500 malware and benign reports from ANY.RUN, we used five big language models to generate 5,000 question-context-answer triples covering malware identification, technical indicators, behavioral patterns, and mitigatienter-labelon strategies. A multilayered validation method, comprising format checks, deduplication, similarity filtering, and LLM-as-Judge review, revealed high dataset quality, with 100% logical validity and relevance. However, 2% of responses indicated minor incompleteness. Performance benchmarking revealed tradeoffs amongst models in terms of cost, speed, and instruction adherence. The generated dataset and pipeline provide the foundation for intelligent forensic tools and reproducible operations. Future work will include fine-tuning LLMs for particular forensic tasks, creating Retrieval-Augmented Generation systems, investigating multi-agent analysis, and expanding the dataset to encompass more platforms and sample sizes.

REFERENCES

[1] Z. Yin, Z. Wang, W. Xu, J. Zhuang, P. Mozumder, A. Smith, and W. Zhang, "Digital forensics in the age of large language models," *null*, 2025.

[2] P. Sharma and L. Awasthi, "Next-generation digital forensics challenges and evidence preservation framework for iot devices," *International Journal of Next-Generation Computing*, 2023.

[3] N. Nelufule, T. Singano, and M. Masango, "A comprehensive exploration of digital forensics investigations in embedded systems, ubiquitous computing, fog computing, and edge computing," *2024 International Conference on Artificial Intelligence, Big Data, Computing and Data Communication Systems (icABCD)*, 2024.

[4] A. W. Malik, D. S. Bhatti, T.-J. Park, H. U. Ishtiaq, J.-C. Ryou, and K.-I. Kim, "Cloud digital forensics: Beyond tools, techniques, and challenges," *Italian National Conference on Sensors*, 2024.

[5] P. Narasimhan and D. N. Kala, "Emerging trends in digital forensics : Investigating cybercrime," *International Journal of Scientific Research in Computer Science Engineering and Information Technology*, 2025.

[6] B. Sharma, J. Ghawaly, K. McCleary, A. Webb, and I. M. Baggili, "Forensicllm: A local large language model for digital forensics," *Digital Investigation. The International Journal of Digital Forensics and Incident Response*, 2025.

[7] A. Zouhri, L. Zitoune, and I. Lahsen-Cherif, "WiFiQnA: a WiFi dataset for large language models," https://hal.science/hal-05146992, 2025, hAL Id: hal-05146992.

[8] M. Bellouch, L. Zitoune, I. Lahsen-Cherif, and V. Vèque, "Pareto DQL-MultiMDP sub-controllers for load balancing in large and dynamic WiFi networks," in *IEEE International Conference on Communications (ICC)*, Montréal, Canada, Jun. 2025, hAL Id: hal-04952832. [Online]. Available: https://hal.science/hal-04952832

[9] M. Bellouch, L. Zitoune, I. Lahsen-Cherif, and V. Vèque, "Load balancing in large wifi networks using dql-multimdp with constrained clustering," in *2024 32nd International Conference on Modeling, Analysis and Simulation of Computer and Telecommunication Systems (MASCOTS)*, 2024, pp. 1–8.

[10] Y. Bai, J. Zhang, X. Lv, L. Zheng, S. Zhu, L. Hou, Y. Dong, J. Tang, and J. Li, "Longwriter: Unleashing 10,000+ word generation from long context llms," *arXiv.org*, 2024.

[11] H. Xin, D. Guo, Z. Shao, Z. Ren, Q. Zhu, B. L. B. Liu), C. Ruan, W. Li, and X. Liang, "Deepseek-prover: Advancing theorem proving in llms through large-scale synthetic data," *arXiv.org*, 2024.

[12] G. Michelet and F. Breitinger, "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*, 2023.

[13] M. Chernyshev, Z. A. Baig, and R. Doss, "Towards large language model (llm) forensics using llm-based invocation log analysis," *LAMPS@CCS*, 2023.

[14] T. M. Ali and P. Kostakos, "Huntgpt: Integrating machine learning-based anomaly detection and explainable ai with large language models (llms)," *arXiv.org*, 2023.

[15] D. Kundu, "Scifaultyqa: Benchmarking llms on faulty science question detection with a gan-inspired approach to synthetic dataset generation," *arXiv.org*, 2024.

[16] N. Tihanyi, M. Ferrag, R. Jain, T. Bisztray, and M. Debbah, "Cybermetric: A benchmark dataset based on retrieval-augmented generation for evaluating llms in cybersecurity knowledge," *Computer Science Symposium in Russia*, 2024.

[17] M. T. Alam, D. Bhusal, L. Nguyen, and N. Rastogi, "Ctibench: A benchmark for evaluating llms in cyber threat intelligence," *Neural Information Processing Systems*, 2024.

[18] J. Zhang, Y. Bai, X. Lv, W. Gu, D. Liu, M. Zou, S. Cao, L. Hou, Y. Dong, L. Feng, and J. Li, "Longcite: Enabling llms to generate fine-grained citations in long-context qa," *arXiv.org*, 2024.

[19] T. Gao, H. Yen, J. Yu, and D. Chen, "Enabling large language models to generate text with citations," *Conference on Empirical Methods in Natural Language Processing*, 2023.

[20] M. Li, J. Chen, L. Chen, and T. Zhou, "Can llms speak for diverse people? tuning llms via debate to generate controllable controversial statements," *Annual Meeting of the Association for Computational Linguistics*, 2024.

[21] T. Liang, C. Jin, L. Wang, W. Fan, C. Xia, K. Chen, and Y. Yin, "Llm-redial: A large-scale dataset for conversational recommender systems created from user behaviors with llms," *Annual Meeting of the Association for Computational Linguistics*, 2024.

[22] S. Yun, H. Lin, R. Thushara, M. Q. Bhat, Y. Wang, Z. Jiang, M. Deng, J. Wang, T. Tao, J. Li, H. Li, P. Nakov, T. Baldwin, Z. Liu, E. P. Xing, X. Liang, and Z. Shen, "Web2code: A large-scale webpage-to-code dataset and evaluation framework for multimodal llms," *Neural Information Processing Systems*, 2024.

[23] K. Chang, K. Wang, N. Yang, Y. Wang, D. Jin, W. Zhu, Z. Chen, C. Li, H. Yan, Y. Zhou, Z. Zhao, Y. Cheng, Y. Pan, Y. Liu, M. Wang, S. Liang, Y. Han, H. Li, and X. Li, "Data is all you need: Finetuning llms for chip design via an automated design-data augmentation framework," *Design Automation Conference*, 2024.