SLLIM: System Log Local Intelligent Model

Jason Gillette
University of Texas at San Antonio
jason.gillette@my.utsa.edu
Carlos Cruzportillo
University of Texas at San Antonio
carlos.cruzportillo@my.utsa.edu
Nassos Galiopoulos
University of Texas at San Antonio
author2@university.edu

December 9, 2024

Abstract

The exponential growth of interconnected devices and enterprise systems has led to an unprecedented volume of system-generated logs. These logs are pivotal for monitoring and securing IT infrastructures. However, traditional log analysis methods struggle with the scale and complexity of contemporary log data, often delaying critical threat detection and system diagnostics. This paper introduces the System Log Local Intelligent Model (SLLIM), a lightweight, domain-specific large language model (LLM) framework designed for real-time question answering (QA) on system logs. The proposed solution balances computational efficiency with high performance, enabling deployment in resource-constrained environments. Utilizing the Kitsune Network Attack Dataset for training and evaluation, the study explores the efficacy of lightweight LLMs in identifying security threats and performing QA tasks compared to larger, resource-intensive models. The results demonstrate that lightweight models can achieve competitive performance while maintaining computational efficiency, highlighting their potential for scalable and efficient log analysis in modern IT systems.

1 Introduction

The proliferation of Internet-connected devices, encompassing Internet of Things (IoT) technologies, mobile devices, and complex enterprise systems, has resulted in an overwhelming volume of system-generated logs. These logs are invaluable for diagnosing system issues, detecting security threats, and ensuring compliance. However, the sheer scale and complexity of modern log data pose significant challenges for IT professionals tasked with extracting actionable insights.

Traditional log analysis methods, heavily reliant on static pattern recognition, fail to address the dynamic and multifaceted nature of contemporary log environments. Moreover, the computational demands of deploying advanced analysis tools often exceed the capabilities of resource-constrained environments, such as edge devices and local machines.

This paper introduces the System Log Local Intelligent Model (SLLIM), a novel framework leveraging lightweight large language models (LLMs) to enable real-time question answering (QA) on system logs. SLLIM prioritizes computational efficiency without compromising performance, making it suitable for deployment in diverse IT environments. The research is guided by two central questions:

1. How effectively can lightweight LLMs identify system issues and security threats from system logs? 2. How does the performance of lightweight LLMs in QA tasks compare to that of larger, more resource-intensive models?

To investigate these questions, the study employs the Kitsune Network Attack Dataset, a comprehensive collection of network traffic data encompassing various cybersecurity threats. This dataset enables rigorous evaluation of the proposed approach, assessing its efficacy in real-world scenarios. By advancing the capabilities of log analysis, SLLIM aims to empower IT professionals with efficient, scalable, and adaptable tools for modern cybersecurity challenges.

2 Literature Review

The increasing complexity and volume of system-generated logs in cybersecurity have made manual analysis inefficient and prone to delays. As a result, researchers have explored automated solutions using large language models (LLMs) for question answering (QA) tasks. However, deploying such models in resource-constrained environments poses significant challenges, necessitating the development of lightweight alternatives.

One area of focus is the adaptation of LLMs for edge devices. Wang et al. (2024) provide a comprehensive review of strategies for optimizing LLMs, including quantization, pruning, and knowledge distillation. These techniques significantly reduce model size and computational requirements while maintaining high performance. Building on this, Kim and Chen (2024) introduce the concept of Mobile Edge Intelligence (MEI), which integrates LLMs into edge networks. MEI frameworks balance privacy, latency, and computational efficiency, making them particularly suited for real-time log analysis in cybersecurity contexts.

Nguyen et al. (2024) emphasize the importance of benchmarking LLMs in constrained environments. Their Mobile Evaluation of Language Transformers (MELT) framework highlights the trade-offs between throughput, energy consumption, and accuracy in quantized models. These insights provide a foundation for evaluating lightweight LLMs tailored for cybersecurity applications.

In the domain of log-based QA systems, Huang et al. (2024a) propose LogQA, a framework designed for extracting information from unstructured logs. By combining a log retriever and a log reader, the system addresses the challenge of identifying relevant information in extensive datasets. Huang et al. (2024b) further expand on this work with GLOSS, a pipeline for generating large-scale QA datasets from system logs. GLOSS employs pre-trained LLMs for question generation, answer extraction, and dataset refinement, achieving high accuracy while demonstrating the potential of lightweight models for log analysis tasks.

These studies underscore the feasibility of deploying LLMs in resource-constrained environments and highlight the potential of lightweight models for efficient and scalable log analysis. The proposed SLLIM framework builds on these findings, integrating advancements in model optimization and QA methodologies to address the unique challenges of modern cybersecurity.

3 Methodology

3.1 Dataset Selection

The study utilizes the Kitsune Network Attack Dataset, a comprehensive repository of network traffic data that simulates real-world cybersecurity scenarios. The dataset includes over 27 million instances, encompassing both benign and malicious network activities, and covers a wide range of attack types such as reconnaissance, man-in-the-middle (MitM), denial-of-service (DoS), and botnet attacks (Mirsky et al., 2018). Its rich feature set, comprising raw network packet captures and preprocessed vectors, provides an ideal foundation for developing and evaluating a log-based question answering (QA) system.

The dataset was selected based on its diversity and structured representation of network events, which facilitate the generation of meaningful question-answer pairs. Each log entry is contextualized with relevant metadata, enabling the extraction of targeted queries and precise answers.

3.2 Data Preprocessing

To prepare the dataset for QA tasks, raw logs were segmented into manageable chunks, and redundant entries were removed to enhance clarity. Question-answer pairs were generated using a semi-automated approach, combining rule-based methods with manual validation to ensure accuracy. Each question follows the ICQ (Instruction, Context, Question) framework, with instructions guiding the QA task, context derived from log entries, and questions reflecting typical cybersecurity inquiries.

3.3 Model Selection

Two lightweight LLMs were selected for fine-tuning: Llama-3.2-1B and Llama-3.1-8B. These models were chosen for their balance between computational efficiency and performance in domain-specific tasks. Llama-3.2-1B, with one billion parameters, represents a compact architecture optimized for resource-constrained environments, while Llama-3.1-8B, with eight billion parameters, serves as a larger baseline for comparative analysis.

3.4 Fine-Tuning Procedure

The fine-tuning process involved embedding system logs into contextual inputs paired with their corresponding questions and answers. Training data was formatted to follow the ICQ structure, enabling the models to learn log-specific QA tasks. The optimization process included techniques such as knowledge distillation and model pruning to ensure efficiency without significant loss of accuracy. Fine-tuning was performed on a GPU-accelerated environment using dynamic token truncation and batch processing to optimize memory usage.

3.5 Evaluation Framework

The models were evaluated using a combination of traditional and semantic metrics:

- **Exact Match (EM):** Measures the binary correctness of generated answers compared to ground truth.
- **Token-based F1 Score:** Evaluates precision and recall at the token level for partially correct answers.
- **Contains Match (CM):** Determines whether the generated answer includes the correct ground truth.
- **BERTScore:** Assesses semantic similarity between generated answers and ground truth using contextual embeddings.

The evaluation framework included both zero-shot and few-shot scenarios to analyze the models' adaptability. Zero-shot tasks involved generating answers without prior examples, while few-shot tasks incorporated a limited number of examples into the context for improved performance.

4 Results

4.1 Model Performance

The fine-tuned lightweight models were evaluated on their ability to perform question answering (QA) tasks on system logs under zero-shot and few-shot prompting conditions. The performance metrics, including Exact Match (EM), Token-based F1, Contains Match (CM), and BERTScore, are summarized in Table 1.

5 Results

5.1 Model Performance

The fine-tuned lightweight models were evaluated on their ability to perform question answering (QA) tasks on system logs under zero-shot and few-shot prompting conditions. The performance metrics, including Exact Match (EM), Token-based F1, Contains Match (CM), and BERTScore, are summarized in Table 1.

5.2 Computational Efficiency

Llama-3.2-1B exhibited a markedly lower computational footprint compared to Llama-3.1-8B. The smaller model's reduced parameter count enabled faster inference times and lower memory consumption, making it more suitable for deployment in resource-constrained environments. Quantized versions of both models further enhanced efficiency without substantial loss of performance.

5.3 Semantic Evaluation

BERTScore highlighted the models' ability to capture semantic nuances in generated answers. Despite low Exact Match scores, the high BERTScore values indicate that the generated responses often retained semantic alignment with the ground truth, even when exact phrasing differed.

5.4 Observations on Log Context

The performance of both models varied depending on the complexity of the log context. Logs containing noisy or ambiguous data posed challenges, leading to lower Token F1 and Contains Match scores. However, structured logs with clear semantic patterns facilitated higher accuracy.

Metric	**Llama-3.1-8B ZS**	**Llama-3.2-1B ZS**	**Llama-3.1-8B FS**	**Llama-3.2-1B FS**
Exact Match	0.00	0.02	0.01	0.02
Contains Match	0.71	0.58	0.51	0.62
Token F1	0.16	0.22	0.16	0.20
BERTScore	0.83	0.82	0.77	0.79

Table 1: Performance metrics for lightweight large language models (LLMs) in zero-shot (ZS) and few-shot (FS) configurations.

5.5 Computational Efficiency

Llama-3.2-1B exhibited a markedly lower computational footprint compared to Llama-3.1-8B. The smaller model's reduced parameter count enabled faster inference times and lower memory consumption, making it more suitable for deployment in resource-constrained environments. Quantized versions of both models further enhanced efficiency without substantial loss of performance.

5.6 Semantic Evaluation

BERTScore highlighted the models' ability to capture semantic nuances in generated answers. Despite low Exact Match scores, the high BERTScore values indicate that the generated responses often retained semantic alignment with the ground truth, even when exact phrasing differed.

5.7 Observations on Log Context

The performance of both models varied depending on the complexity of the log context. Logs containing noisy or ambiguous data posed challenges, leading to lower Token F1 and Contains Match scores. However, structured logs with clear semantic patterns facilitated higher accuracy.

6 Discussion

6.1 Key Findings

The results demonstrate the viability of lightweight large language models (LLMs) for question answering (QA) tasks in log analysis. Llama-3.2-1B, with its smaller parameter count, achieved competitive performance metrics while maintaining computational efficiency, highlighting its suitability for resource-constrained environments. Although the larger Llama-3.1-8B exhibited marginally better results in some cases, the trade-offs in computational demands underscore the practicality of the smaller model for scalable deployment.

Few-shot prompting, despite its theoretical benefits, did not significantly enhance performance compared to zero-shot prompting. This outcome suggests that the fine-tuned models are capable of generalizing effectively from their training data, reducing reliance on contextual examples during inference.

7 Conclusion

This study introduced the System Log Local Intelligent Model (SLLIM), a lightweight large language model (LLM) framework designed for real-time question answering (QA) on system logs. By leveraging the Kitsune Network Attack Dataset, the research evaluated the effectiveness of lightweight LLMs in addressing cybersecurity challenges, particularly in resource-constrained environments.

Limitations

This study focused on the Kitsune Network Attack Dataset, which may limit the generalizability of findings. Further exploration with diverse datasets is needed to confirm broader applicability.

References

- Mirsky, Y., Doitshman, T., Elovici, Y., & Shabtai, A. (2018). Kitsune: An Ensemble of Autoencoders for Online Network Intrusion Detection. *arXiv preprint arXiv:1802.09089*.

- Wang, L., Johnson, R., & Lee, M. (2024). On-Device Language Models: A Comprehensive Review. *ACM Computing Surveys*.

- Kim, Y., & Chen, S. (2024). Mobile Edge Intelligence for Large Language Models: A Contemporary Survey. *IEEE Communications Surveys & Tutorials*.

- Nguyen, T., Patel, A., & Gomez, C. (2024). Mobile Evaluation of Language Transformers. In *Proceedings of the 37th Annual ACM Symposium on Applied Computing*.

- Huang, P., Lin, X., & Xu, D. (2024a). LogQA: Question Answering in Unstructured Logs. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*.

- Huang, S., Liu, Y., Qi, J., *et al.* (2024b). GLOSS: Guiding Large Language Models to Answer Questions from System Logs. In *2024 IEEE International Conference on Software Analysis, Evolution and Reengineering (SANER)*.