Leveraging Large Language Models for Log Anomaly Detection: Case Studies and Comparative Analysis

Yujing Xiong *

School of Computer Science and Engineering, University of Electronic Science and Technology of China, Chengdu, Sichuan, 610000, China

Abstract. As computer systems become more complex, log anomaly detection has become more crucial and challenging. However, the conventional methods' dependence on log parsing places significant limitations on them. Many problems still beset the widely used deep-learning techniques. Large Language Models (LLMs) offer an innovative approach to these restrictions via their advanced contextual reasoning and natural language comprehension skills. The enormous potential of LLMs to improve performance from an abundance of aspects throughout the various log anomaly detection stages will serve as the primary focus of this study. Several frameworks are examined in the case study, applying LLMs to improve the frameworks' performance during the anomaly detection stage as well as to construct the preprocessed data. The study will additionally explore how some LLM-related technologies, for example, Retrieval-Augmented Generation (RAG), integrate into this field. In the meantime, the benefits and drawbacks of these frameworks are also discussed. The study will demonstrate the tremendously vast potential of LLM in log anomaly detection through the comparative analysis. Finally, this paper discusses the current challenges and limitations of LLM-based approaches and proposes directions for future development.

1 Introduction

System log data records important runtime conditions and operational details while the system is being monitored, which serves as the most basic way for humans to understand system behaviors. In these logs, anomalous entries often contain vital signals about abnormal operation. Hence, such data can be used to identify abnormal activities or supporting tasks such as security analysis and troubleshooting. Therefore, log anomaly detection has emerged as a significant concern in system management and cybersecurity.

Conventional log anomaly detection techniques primarily use manually created regular expression templates to parse logs and convert unstructured log data into structured data for analysis [1]. However, a vast amount of logs are created as computer systems and applications advance increasingly sophisticated, making the conventional approaches

* Corresponding author: 2023310101020@std.uestc.edu.cn

laborious and ineffective. Therefore, automatic log parsing is highly needed for improving efficiency. For example, a parse tree that has a fixed depth was deployed in Drain to transform logs into various templates in real-time. The same idea of automatic parsing is also applied in Spell, which uses a longest common subsequence (LCS) approach for parsing [2-3].

In recent years, deep learning, which has advanced significantly, exhibits great future potential in log anomaly detection. For instance, Long Short-term Memory (LSTM) has demonstrated remarkable outcomes in log anomaly detection [4]. Log Robust is also an approach that applies such an idea. It leverages an attention-based Bi-LSTM model to capture contextual patterns and weigh event significance autonomously. This approach demonstrated a 13.2% improvement in F1-score over traditional methods on the HDFS dataset under conditions where log events evolve, achieving an F1 of 0.963 [5].

To overcome the limitations mentioned above, LLMs have attracted more and more attention. Some research concentrated on applying LLMs for log template extraction in view of the importance of extracting the template from log entries. For instance, Xu et al. proposed DivLog, which achieves 98.1% parsing accuracy on the HDFS dataset via prompt-enhanced in-context learning to guide LLMs in extracting accurate templates and choosing representative demonstrations [6]. Along with template extraction, LLMs show a lot of promise in the anomaly detection phase. LogGPT deploys generative pre-trained transformers to identify anomalies and model sequential log patterns without the need for manually constructed rules [7]. In conclusion, LLMs provide a flexible substitute for conventional methods for generalization across a variety of highly complicated log sources. There are still challenges, though, like generalization and model interpretability.

This review explores how LLMs can be leveraged in detecting abnormal logs, highlighting their strengths in both template extraction and anomaly detection. This research aims to investigate how LLMs can enhance the performance of detecting abnormal logs and the distinctive effects resulting from their integration with different technologies.

2 Background

2.1 Log parsing

Unstructured textual data, including both dynamic variables (such as IP addresses and IDs) and immutable constant elements, is commonly found in raw system logs. One of the most important preprocessing steps in the log analysis pipeline is log parsing. In order to transform this unstructured data into structured formats and make the ensuing log analysis processes easier and more effective, the goal of log parsing is to separate variable parameters and extract log templates from unstructured log messages.

2.2 Deep learning-based anomaly detection

Deep learning has developed as the dominant paradigm for log anomaly detection in recent years, primarily through two approaches: supervised binary classification and semi-supervised sequence generation. Supervised methods like LogRobust use embedding techniques and Bi-LSTM networks to classify anomalies, while semi-supervised methods like DeepLog and NeuralLog model normal log sequences using LSTMs or BERT-based transformers to detect deviations [5, 8, 9].

2.3 LLMs for anomaly log detection

LLMs are increasingly prevalent in recent years and have made tremendous achievements in many fields. Accordingly, by utilizing their deep semantic understanding and contextual reasoning skills, advancements in LLMs have drastically changed log-based anomaly detection [6-7]. In contrast to conventional deep learning techniques, LLMs are highly effective at handling unstructured log data without the need for extensive feature engineering or predefined templates. The main limitations of the methods mentioned above can be effectively addressed by their capability to understand different patterns of logs and generate explanations that are highly comprehensible to humans.

3 Case studies

3.1 Case study 1: Utilizing large language models as a template parser and anomaly detector

An essential part of previous methods for detecting anomalous logs is log parsing. However, there are two major problems with these methods. Firstly, when handling complex log formats, current log parsers frequently have limited accuracy. Second, sequence-based detection models are not flexible enough to adjust to evolving log patterns. The advantages of LLMs present an opportunity to get past these obstacles. Their strong semantic comprehension skills may avoid the loss of important semantic information that often happens in conventional parsers and significantly boost log parsing accuracy.

The framework developed by Zhou et al. states that an LLM can serve as a parser to convert logs into structured data, and that BERT-based intelligent analysis is used in the log anomaly detection phase that follows [10]. More specifically, ChatGPT is applied to acquire log templates during the beginning log parsing stage. The framework determines the pattern of normal log sequences in its following log anomaly detection stage, where it employs the BERT-based LogBERT model. The framework efficiently maintains the semantic information while improving log parsing performance.

The experimental results showed that this framework performed extremely effectively on public log datasets like HDFS and BGL. On the BGL datasets, it specifically obtained an F1-score of 90.8%, a recall rate of 92.3%, and a precision of 89.4%. This result significantly outperforms many traditional parser-based deep learning techniques, demonstrating the effectiveness of LLMs in improving log parsing quality and demonstrating the benefit of combining them with downstream detection models.

3.2 Case study 2: A semi-supervised approach to log anomaly detection using a large language model enhanced by RAG

Similar to the limitations mentioned earlier, the traditional methods of log anomaly detection are highly limited in log parsing and sequence modeling. Due to the errors caused by the incorrect parsing and the various log patterns that are still evolving these days, these conventional approaches have limited precision and generalization. RAG, with its powerful knowledge retrieval and semantic understanding, holds great potential to improve the precision and its interpretability when it is combined with LLMs.

LogRAG, the framework proposed by Zhang et al., effectively reduces the risk of log parsing errors and limitations of sequence modeling [11]. In order to extract log templates and log tokens independently, this method first uses a novel feature extraction strategy in the log preprocessing stage. The approach uses OpenAI's text-embedding-ada-002 for feature

representation from the log templates and BigLog, a pretrained model, to encode log tokens during the log presentation stage in order to preserve semantic information. A semi-supervised anomaly detection model built on a one-class classifier algorithm is then employed to find anomalous logs during the anomaly detection phase. These logs undergo additional processing because some normal logs that have never been examined before may be incorrectly labeled as abnormal. The most novel feature is the RAG-enhanced post-processing step, which verifies again logs that were first deemed abnormal by using RAG.

The LogRAG framework exhibits significant advantages on the BGL and Spirit datasets. On the BGL dataset, LogRAG-Mistral achieved an F1 score of 0.898 with a window size of 200, surpassing all semi-supervised baseline methods. And the best baseline, PLELog, achieved an F1 score of 0.752, representing a performance improvement of 19.4%. On the Spirit dataset, LogRAG-ChatGPT achieved an F1 score of 0.927 with a 0.5-hour window setting, demonstrating superior detection performance compared to the best semi-supervised baseline PLELog, which achieved an F1 score of 0.581, representing a 59.6% improvement.

3.3 Case study 3: A three-stage framework achieving superb performance and high interpretability in log anomaly detection

Although LLMs tend to be extremely powerful at understanding natural language, their lack of background knowledge in cybersecurity causes them to have a much higher rate of false positives when processing multi-threaded logs. Furthermore, issues with high computational costs and the inherent class imbalance of log data make full parameter fine-tuning difficult.

Therefore, Zhang et al. proposed the LLM-LADE framework [12]. The framework innovatively designs a novel three-stage architecture. Firstly, the methodology constructed a seed dataset by manually annotating the anomaly reasons for a small subset of log sequences by domain experts. Then ChatGPT, with its powerful reasoning capabilities, was leveraged for large-scale data expansion. LoRA was used to fine-tune LLaMA3-8B during the Fine-Tuning phase. The framework utilizes a dynamically weighted loss function to improve the model's attention to sparse anomaly samples. Lastly, a continuous learning mechanism guided by experienced human feedback is implemented in the Online Optimization stage.

The LLM-LADE framework achieves outstanding detection performance on three datasets (BGL, Thunderbird, and HDFS) and outperforms all baseline models in the experiment section with F1-scores of 0.990, 0.989, and 0.997, respectively. It is particularly noteworthy that the framework significantly improves the interpretability of outcome data. Through dual validation involving automated GPT-4 evaluation and manual assessment by domain experts, the generated explanations received average scores of 4 out of 5 for both usefulness and readability.

3.4 Comparative Analysis

The lack of semantic comprehension caused by a heavy reliance on log parsing and related problems in sequence modeling is overcome in all three of the aforementioned studies by the significant natural understanding ability of LLMs. Table 1 summarizes the distinctive technologies they applied.

Table 1. Comparative Analysis of Solutions to Core Challenges.

Framework	Approach to Mitigating Semantic Information Loss	Approach to Temporal Sequence Modeling
LogBERT (3.1)	Employs LLMs for log parsing.	Utilizes BERT's Self-Attention mechanism coupled with an MLM pretraining task.

LogRAG (3.2)	Adopts the RAG approach and integrates context from both log tokens and log templates.	Employs a one-class classifier
LLM-LADE (3.3)	Directly leverages a fine-tuned LLM for detection and judgment.	Relies on the inherent sequence modeling capabilities of the LLM itself.

Furthermore, LLMs play a variety of roles within each framework. LLMs can contribute to data preprocessing, post-processing, and the critical anomaly detection phase due to their semantic capabilities and flexibility in response to the development of log systems. The precise function of LLMs and related goals within the previously mentioned frameworks are displayed in Table 2.

Table 2. Comparative Analysis of Solutions to Core Challenges.

Framework	**LLM Application Stage**	**Integration Methodology**	**Objective**
LogBERT (3.1)	Pre-Processing	Prompt Engineering （Few-shot）	To replace traditional parsers and generate more accurate log templates.
LogRAG (3.2)	Post-Processing	Prompt Engineering +RAG	To validate the abnormal logs, reduce false positives, and provide reasoning for the decision.
LLM-LADE (3.3)	Data Augmentation	Prompt Engineering （Few-shot）	To automatically generate training data for the fine-tuning model.
	Core Detection & Explanation	PEFT LoRA	To perform anomaly detection and generate explanations.

The performance and architectural merits of the models mentioned above are also different. Table 3 summarizes the main advantages and drawbacks of each distinct framework by performing a general cross-sectional comparison of the three approaches based on their performance on the BGL dataset.

Table 3. Comparative Analysis of Solutions to Core Challenges.

Framework	**Performance on BGL**			**Core Advantages**	**Limitations**
	Prec	**Rec**	**F1**		
LogBERT	89.4	92.3	90.8	Enhanced Parsing Accuracy	Disregards critical information (such as timestamps, security level, and so on)
LogRAG-ChatGPT (window=0.5h)	0.957	0.828	0.888	Significantly Reduced False Positive Rate	The core deep single-classifier model needs periodic retraining.
LLM-LADE	0.992	0.987	0.989	Exceptional Performance Interpretability Online Continuous Learning Capability	High computational cost Unproven cross-domain generalizability

According to the comparison, although LLMs are integrating into each framework, LLM-LADE outperforms the rest of the frameworks, which can be ascribed to its advanced architecture design and comprehensive consideration of practical deployment issues. Furthermore, a key observation is that the core advantages of these frameworks are linked to the inherent properties introduced by the LLMs themselves, thereby strongly validating the immense potential of LLMs in the field of anomaly log detection.

4 Discussion

4.1 Challenges and limitations

4.1.1 Model hallucination and knowledge gaps

Despite the fact that LLMs have tremendous general knowledge, the pre-training data still contain insufficient specific domain knowledge related to particular IT systems, cybersecurity, and fault patterns. This drawback may lead to model hallucinations, which stands for the generation of seemingly plausible but factually incorrect explanations or judgments [13]. Utilizing prompt engineering is far from solving this challenge since the LLM, due to its inherent properties, lacks a profound understanding of the specific systems within that domain.

4.1.2 Generalization capability

The datasets currently used in most research on log anomaly detection, such as BGL, HDFS, and Thunderbird, originate from real system logs, which leads to covering limited log types, primarily focusing on High-Performance Computing (HPC) and distributed systems. Consequently, the architecture lacks the capability of generalizing to complex, dynamic, multi-source logs. The models' performance is likely to suffer when they are implemented in new scenarios with domains that are very different from the training data. This problem demonstrates how current approaches overfit to particular datasets and lack general proficiency with universal log semantics.

4.1.3 Interpretability

LLMs themselves are black-box models and thus hinder us from validating their internal decision process. For practical, real-world systems, system operators place a high value on understanding the underlying causes of log anomalies. LLMs' limited explainability is what renders it difficult for them to provide maintenance engineers with the proper assistance they need when maintaining and troubleshooting systems.

4.2 Future directions

4.2.1 Enhance domain adaptation capability

Future research should not solely concentrate on full-parameter fine-tuning or prompt strategies due to the aforementioned difficulties. The issue of a lack of adaptability can be addressed with methods like RAG that improve domain adaptation [11]. In the meantime, frameworks can also give priority to online continuous learning capabilities, which will improve their ability to adapt to a variety of domain systems and produce accurate judgments.

4.2.2 Enhance explainability

The subsequent studies are highly needed to concentrate more on developing an explainable framework and enhancing the model's loyalty because the explainability of log systems is extremely important. In the meantime, creating a more rational assessment can enhance interpretability. A human-in-the-loop mechanism where professionals assess and rectify the

LLM's initial outputs could be a promising strategy [12]. The feedback can be utilized to improve the model's explainability or update existing knowledge bases.

5 Conclusion

An overview of the application of LLMs to anomaly log detection is given in this paper. Traditional techniques primarily depend on log parsing, yet they frequently fail to detect patterns in logs. Furthermore, despite their recent popularity, deep learning-based approaches have limited adaptability and depend heavily on data. Thus, LLMs' sophisticated natural language processing and semantic understanding skills are becoming steadily more crucial. The case study illustrates how successfully LLMs perform anomaly log detection throughout multiple pipeline stages. LLMs can successfully take the place of traditional parsers in the data preprocessing step when it comes to extracting log patterns. In the post-processing stage, combining LLMs with RAG technology can significantly lower the rate of false positives. Overall, LLMs hold tremendous potential to improve the performance of various stages of frameworks for analyzing anomaly logs. The study also notes that these LLM-based frameworks leverage their natural language understanding to address the loss of semantic information during log parsing. Different architectures also adopt varied approaches to sequence modeling, either preserving contextual understanding for more accurate judgments or employing a single classifier to avoid misclassifying logs from different threads as anomalies. Nevertheless, current research also encounters some challenges, such as knowledge gaps, limited generalization, and poor interpretability, hindering practical application. Future research should prioritize enhancing adaptability and interpretability to further solidify the impact and utility of LLMs in anomaly log analysis.

References

1. T. Zhang, H. Qiu, G. Castellano, M. Rifai, C. S. Chen, F. Pianese, System log parsing: A survey. IEEE Trans. Knowl. Data. Eng. **35**, 8596-8614 (2023)
2. M. Du, F. Li, Spell: Streaming parsing of system event logs. In Proceedings of the 2016 IEEE 16th International Conference on Data Mining (ICDM), Barcelona, Spain, December 12-15 (2016), 859-864
3. P. He, J. Zhu, Z. Zheng, M. R. Lyu, Drain: An online log parsing approach with fixed depth tree. in Proceedings of the 2017 IEEE International Conference on Web Services (ICWS), Honolulu, HI, USA, June 25-30 (2017), 33–40
4. R. Vinayakumar, K. P. Soman, P. Poornachandran, Long short-term memory based operation log anomaly detection, in Proceedings of the 2017 International Conference on Advances in Computing, Communications and Informatics (ICACCI), Udupi, India, September 13–16 (2017), 236–242
5. X. Zhang, Y. Xu, Q. Lin, B. Qiao, H. Zhang, Y. Dang, … D. Zhang, Robust log-based anomaly detection on unstable log data. in Proceedings of the 2019 27th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering, Tallinn, Estonia, August 26–30 (2019), 807–817
6. J. Xu, R. Yang, Y. Huo, C. Zhang, P. He, Divlog: Log parsing with prompt enhanced in-context learning. in Proceedings of the IEEE/ACM 46th International Conference on Software Engineering, Lisbon, Portugal, April 14-20 (2024), 1–12
7. J. Qi, S. Huang, Z. Luan, S. Yang, C. Fung, H. Yang, … Z. Wu, Loggpt: Exploring chatgpt for log-based anomaly detection. in Proceedings of the 2023 IEEE International Conference on High Performance Computing & Communications, Data Science & Systems, Smart City & Dependability in Sensor, Cloud & Big Data Systems

& Application (HPCC/DSS/SmartCity/DependSys), Melbourne, Australia, December 17-21 (2023), 273–280
8. M. Du, F. Li, G. Zheng, V. Srikumar, Deeplog: Anomaly detection and diagnosis from system logs through deep learning. in Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security, Dallas, TX, USA, October 30-November 3 (2017), 1285–1298
9. Z. Chen, Q. Gao, L. S. Moss, NeuralLog: Natural language inference with joint neural and logical reasoning. in Proceedings of the SEM 2021: The Tenth Joint Conference on Lexical and Computational Semantics, Online, August 5-6 (2021), 78–88
10. Y. Zhou, Y. Chen, X. Rao, Y. Zhou, Y. Li, C. Hu, Leveraging large language models and BERT for log parsing and anomaly detection. Mathematics **12**, 2758 (2024)
11. W. Zhang, Q. Zhang, E. Yu, Y. Ren, Y. Meng, M. Qiu, J. Wang, Leveraging RAG-Enhanced Large Language Model for Semi-Supervised Log Anomaly Detection. in Proceedings of the 2024 IEEE 35th International Symposium on Software Reliability Engineering (ISSRE), Tsukuba, Japan, October 28-31 (2024), 168–179
12. Z. Zhang, S. Li, L. Zhang, J. Ye, C. Hu, L. Yan, LLM-LADE: Large language model-based log anomaly detection with explanation. K. B. S. **326**, 114064 (2025)
13. Z. Ji, N. Lee, R. Frieske, T. Yu, D. Su, Y. Xu, … P. Fung, Survey of hallucination in natural language generation. ACM Comput. Surv. **55**, 1–38 (2023)