Journal of Optimization of Soft Computing (JOSC)

Vol. 1, Issue 2, pp: (18-25), December-2023

Journal homepage: https://sanad.iau.ir/en/Journal/josc

Research paper

Improved analysis of LUG file-related bulk data using LLG

Azin Pishdad[1], Babak Nikmard[1*], Golnaz Aghaee Ghazvini[1] and Mehrdad Abbasi[1]

1. Department of Computer Engineering- Dolatabad Branch, Islamic Azad University, Dolatabad, Iran

Article Info

Article History:
Received: 2024/1/15
Revised:
Accepted: 2024/2/28

Keywords:
neural network, generative artificial intelligence, large language model, LLM, Log File.

**Corresponding Author's Email Address: a.pishdad@iauda.ac.ir*

Abstract

Nowadays, organizations generate a significant volume of log files that require processing for condition checking, debugging, and anomaly resolution. Outsourcing such processing is not suitable due to the need for real-time processing and security maintenance. Given the multitude of different software and services, organizations face a substantial volume of production logs that should be processed rather than deleted or ignored. In the traditional approach, experts manually check the logs daily. This, on one hand, slows down the process, increases the time and inaccuracy, and, on the other hand, results in a high hiring cost due to the need for an expert force. This article introduces a solution that employs generative neural networks to establish a local structure for log analysis within the organization. The process involves retrieving and parsing text files from various sectors, segmenting them into manageable portions, embedding them, and storing them in a vector database. In this structure, a trained individual without special expertise can quickly access necessary information using appropriate prompts from a local language model available at any time. Therefore, the proposed method can increase the stability of security, increase the speed of analysis, and reduce the costs of human resources.

1. Introduction

As the digital age continues to grant unprecedented access to information, the demand for efficient methods to navigate, search, and extract pertinent data from logs has grown substantially. Organizations generate an enormous volume of logs daily, necessitating thorough analysis. Initially, these files are organized and dispatched to designated individuals responsible for data mining. These individuals are tasked with continuous monitoring of the logs to detect anomalies or suspicious activities. It is at this juncture that the imperative for an automated and streamlined log analysis system becomes evident, rendering the previously employed manual search and interpretation methods obsolete in light of the escalating log volumes.

Typically, an organization's logs are of great value, with a paramount focus on security due to the presence of sensitive information. Consequently, outsourcing such unregulated resources poses the risk of data leakage. The establishment of an in-house system for log analysis and evaluation emerges as the preferred option. This issue finds resolution through the application of artificial intelligence and machine learning tools. In the context of neural networks, intelligent systems, trained with specific parameters, exhibit the capability to comprehend, recover, and furnish meaningful analyses based on a given set of logs.

In a broader sense, the creation of an internal structure using large language models powered by neural networks enables effective data management and user query responses.

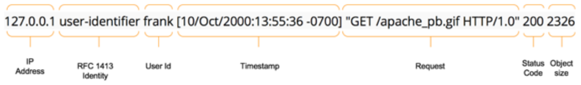

Figure 1- Anatomy of a log file

1.1. Log Management

Logs encompass a collection of data derived from the activities and performance of programs, systems, and users, curated for the purpose of identifying suspicious events. In many business settings, these valuable resources manifest in various formats, including structured, semi-structured, and unstructured files [1]. Log files typically contain sensitive information, and the removal of these records from an organization's internal system can jeopardize data security. The constituents of a log file, akin to its anatomy like figure 1, comprise critical details such as user IDs, system IDs, executed operations with timestamps, information events, and other relevant data [2].

Logs come in various types, including event logs, server logs, access logs, multi-user system logs, resource logs, security logs, and threat logs. These log types are categorized at different levels depending on the ecosystem, such as emergency, warning, informational, and debugging [3]. The management and analysis of logs are instrumental in monitoring the overall performance of programs and tools. By detecting adverse effects and damages, this process effectively distinguishes unexpected and potentially harmful activities, such as errors and intrusions, from routine procedures like the initiation and termination of processes. Additionally, it provides valuable insights to the organization. Handling extensive log data poses a considerable challenge for organizations seeking to monitor their infrastructure for security reasons while maintaining localized control. Various software and systems, such as MongoDB, Kafka, as well as databases, big data, cloud systems, and syslog computing systems, can receive logs. Given the impracticality of outsourcing log management and the unique internal conditions of each organization, the workload of the IT department has increased, necessitating the development and integration of automation tools. Many organizations store their reports within the Security Information and Event Management (SIEM) department. However, as log volumes escalate, this approach becomes problematic, prompting the need for Centralized Log Management (CLM). In a CLM system, data is integrated and directed to a data highway, where it undergoes filtration before being delivered to the intended destination [3]. The implementation of such systems can be facilitated through pipeline creation, ensuring compatibility with large language models and integration into existing infrastructure.

1.2. Analysis of Logs

When dealing with unstructured log files, preprocessing becomes a necessity. To address this, the research conducted by [1] suggests three primary approaches: first: Tokenization: This straightforward method involves segmenting log messages into tokens. It doesn't require a parser and lacks the ability to provide semantic interpretations of symbols. second: Parsing Messages: Parsing involves extracting information from sets of events, encompassing sequences, counts, or statistics. Third: Parameter Extraction: This approach focuses on retrieving parameters, including timestamps, from parsed events.

The analysis process entails the examination and interpretation of logs generated by network systems, operating systems, applications, servers, hardware, and software components. This process provides visibility into the performance and health of the IT infrastructure and application stacks, allowing the detection of potential issues, including security vulnerabilities and failures. It also aids in identifying and preventing problems like overprovisioning and underprovisioning. These techniques are applied to segment documents into manageable chunks, embedding them, and storing them in a vector database to facilitate feeding the document data into a large language model.

Historically, log analysis was performed manually, with experts interpreting log files using text processing software like sedawk. However, the proliferation of logs generated by modern programs has rendered manual analysis infeasible, necessitating automated mechanisms [4]. Hand-coded rules that search for specific keywords have limited utility and are ill-suited for scenarios with unknown parameters [5].

Consequently, the need to employ alternative methods developed by deep learning models has emerged. These methods include pattern recognition using machine learning, keyword tagging, classification, and correlation techniques. Notable examples include the semantic multiphase matching algorithm based on a vector space model and natural language processing (NLP) for matching models [6], as well as the technique for discovering a set of related activities based on k-means clustering in [7] and [8]. Time series analysis is used to detect changes in event frequency [9]. The field of NLP also finds application, such as in linguistic reconstruction [10], where refactoring technologies used in software engineering are applied for syntactic, semantic, and functional reconstruction. Various tools, such as Logstash, ELK Stack, and Sentry, prove helpful.

One of the most significant advantages of log analysis is anomaly detection [1], which involves reporting deviations from normal system behavior by creating models. These approaches may use unsupervised learning, which does not necessitate labeled logs, or supervised learning. Anomaly detection is categorized into two main types: offline and online. In offline anomaly detection systems, notable examples include PCA [11], which utilizes principal component analysis to detect anomalies, and LogCluster [12], which assigns weights to events and employs cumulative hierarchical clustering to identify patterns. Invariants Mining [13] analyzes single values in reports to learn variables, while LogRobust [14] utilizes a Bi-LSTM neural network for learning. Online anomaly detection systems employ supervised learning, using patterns learned offline from normal logs to detect anomalous events in production. Prominent examples include DeepLog [15] and LogAnomaly [16].

This article delves into issues and their resolutions within the realm of daily log analysis, organized into seven sections: The initial section provides an introduction and discusses the current conditions. The second section outlines the artificial intelligence process. Moving forward, the third section addresses the challenges that require solutions. Subsequently, the fourth and fifth sections delineate the proposed solution and its evaluation. The sixth section establishes limitations and outlines the necessary requirements. Finally, the seventh section conclussion the article.

2. Generative Artificial Intelligence

Generative artificial intelligence (GAI) aims to develop machines capable of reasoning and acting in a manner resembling human cognition [17]. This technology empowers AI systems to generate text, images, or other media items in response to user requests, revolutionizing various industries. GAI models acquire an understanding of patterns and structures within their training data and subsequently generate new data. These models can be trained on extensive datasets, enabling multi-stage learning and domain knowledge development through reinforcement learning from human feedback. GAI can be categorized as either unimodal or multimodal. Unimodal systems process a single type of input, while multimodal systems can handle multiple input modalities.

GAI can further be classified into two main categories: task-specific GAI and general GAI. The focus of this research is on investigating general GAI [18]. Advances in big data representation technologies have led to the emergence of a human-interpretable language for patterns and structures within input data, enabling the accomplishment of diverse objectives across various environments. The objective is to transcend the current language generation paradigm, which often involves fitting sample distributions to specific tasks. However, the development of general GAI is still confronted by several key challenges. These challenges include high training and maintenance costs, dispersion of high-quality data, integration of domain knowledge, interpretability, model validity, resource allocation, and security. A comprehensive review of the development of GAI from 2018 to the present is presented in Figure 2 [18], illustrating the evolution of this field.

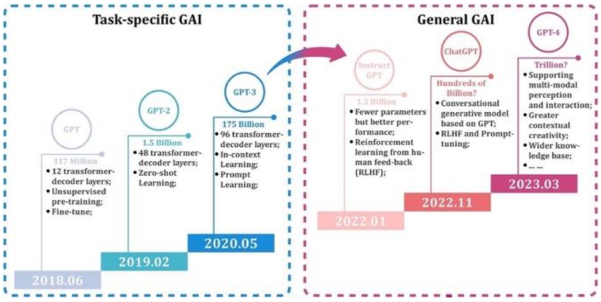

Figure 2. Evolution of GAI [18]

2.1. The Evolution of Generative AI Models

Initially, GPT (Generative Pre-trained Transformer) demonstrated its potential to generate specific natural language through unsupervised pre-training and fine-tuning in downstream tasks. It utilizes 12 transformer decoder layers to perform next-word prediction and generate coherent text. The subsequent model, GPT-2 [19], extended this structure and introduced the concept of zero-shot learning.

Building upon these advances, GPT-3 [20] introduced the use of prompts to reduce the reliance on supervised datasets. It leverages the prior knowledge acquired during the pre-training phase to enhance the quality of generated content. This approach enables the language model to adapt quickly to new scenarios, even with limited or unlabeled data. Additionally, the emergence of reinforcement learning became notable during this period.

InstructGPT, introduced as a general GAI, utilizes Reinforcement Learning from Human Feedback (RLHF) to make decisions aligned with human preferences, achieving improved performance with fewer parameters. This results in an intelligent system characterized by consistency, interpretability, and reliability. RLHF optimizes the original problem by transforming indistinguishable objectives in language production tasks into sequential decision processes [21].

In November 2022, ChatGPT, a general-purpose OpenAI-based chatbot known for its ability to "produce human-like text," attracted millions of users [22]. ChatGPT, built on the foundations of large language models, serves as a conversational user interface and application programming interface (API). It empowers AI applications to generate text and images [17]. These large language models, equipped with self-supervised learning capabilities, exhibit exceptional proficiency in complex tasks and question answering, even in the absence of intrinsic motivations or goals. Beyond language mastery, transformer-based Large Language Models (LLMs), like those in ChatGPT, can tackle diverse and challenging tasks, including mathematics, coding, medical analysis, law, psychology, and more. These models learn from large text datasets through unsupervised learning methods, allowing them to recognize statistical patterns and regularities in human language, thereby generating coherent and contextually relevant responses to user inputs [23]. ChatGPT relies on a self-generated LLM, a machine learning system capable of independent learning from data to produce sophisticated and seemingly intelligent text after extensive training on large text datasets [24].To further enhance the model's capabilities, reinforcement learning from human feedback is integrated with GAI models, enabling domain knowledge to play a role in the training process. This integration marks a shift from "fitting-generation" to "Pretraining-Prompting-Generation" [25].

Recently, GPT-4, the latest model developed by OpenAI, has been trained on an unprecedented scale of computation and data, delivering remarkably human-like performance across a wide array of tasks. The introduction of GPT-4 signifies the evolution of general GAI based on the GPT series, which now includes the understanding and generation of multimodal data, encompassing text, images, and audio. Moreover, it can identify logical errors in reasoning to generate valuable responses [26].

3. Challenges in Log Analysis

The purpose of this article is to investigate three critical aspects: security, time efficiency, and cost reduction in the analysis of logs received from various departments within an organization.

Security:

Logs contain intricate details of valuable and private information pertaining to the organizational structure. Careful examination of this data is crucial, and outsourcing beyond the organization's internal network is not considered. The proposed method establishes a local structure within the organization to uphold the security of such sensitive information.

Time Efficiency:

The sheer volume of logs generated and stored for recovery and analysis is extensive. Traditional methods of checking logs involve writing various queries to extract useful information and reports, which is time-consuming. This results in a slowed-down log checking process, making it nearly impossible to review all logs. The proposed method streamlines and accelerates this process, providing a solution to the time constraints associated with log analysis.

Cost Reduction:

Log analysis typically requires the expertise of specialized individuals, and the annual human resources costs escalate when hiring such personnel. Furthermore, the nature of the job demands more than one person, leading to increased salary expenses. In contrast, the proposed method operates efficiently by reducing employment costs. It achieves this by utilizing a trained professional, eliminating the need for high-level experts and minimizing financial burdens.

4. Proposed Method: Development of report analysis using local LLMs

Development of Report Analysis using Local Large Language Models (LLMs): There exists a diverse array of AI applications, primarily relying on machine learning, deep learning, and natural language processing (NLP). Machine learning harnesses advanced algorithms to analyze data and identify patterns for predictive purposes. Deep learning employs neural networks to process vast datasets. Large language models (LLMs) [20], [32], [33], considered a groundbreaking technology in the realm of natural language processing, empower developers to create previously unattainable applications, such as the

Copilot programming assistant [34]. However, the true potential of these LLMs is realized when they are integrated with other computational or knowledge resources. This amalgamation can result in the creation of customized solutions for individual organizations. With appropriate training, these integrated systems can ingest all available logs and provide meaningful analyses. For instance, organizations like Google have introduced competitive artificial intelligence chatbots like ChatGPT and other advanced models proposed by OpenAI. Google's AI chatbot, known as "Bard," showcases impressive conversational skills. Meanwhile, several Chinese companies, including HUAWEI, Baidu, Alibaba, and Tencent, have offered their LLMs to develop industrial-grade models, contributing to the industrialization of large-scale artificial intelligence models.

To respond to user inquiries regarding the source of collected logs, it's essential to train LLMs. While LLMs possess remarkable capabilities, they lack knowledge beyond what they've been trained on. "Retrieval Augmented Generation" is a technique that enables the creation of intelligent systems connecting a language model to additional data sources to provide instructions to the LLM, enabling it to manage and interact with its environment.

In this proposed method, as illustrated in Figure 3, the log files undergo initial loading and are subsequently divided into smaller, manageable segments using a text splitter to ensure compatibility with the model. These logs are then embedded for storage. Utilizing these embeddings, semantically similar logs to a given query can be retrieved. These snippets are indexed in a database to facilitate future searches and retrievals. This created index can be employed for generation, essentially functioning as a search format that operates in two stages: capturing the logs in a query format, followed by retrieving the appended production chain. A vector database is integrated to construct high-performance vector search programs, enhancing the speed and accuracy of search and retrieval processes. Various index types are available, with Vectorstore being the most common. Its user-friendly API, scalability, and advanced algorithms allow developers to efficiently manage extensive vector data, achieving real-time retrieval and building effective search engines.

When a user submits a query, the query is initially embedded, and a similarity search is conducted in the vector database. The retrieved documents, in conjunction with the query, serve as input to the query chain. This consolidated input is then forwarded to the LLM, enabling it to generate a response that precisely addresses the user's inquiry. The search encompasses the logs relevant to the question and employs both a PromptValue, which acts as model input, and a PromptTemplate, responsible for creating these inputs. PromptValue is conveyed to the model, and the outcome, representing an accurate and well-informed response based on the information found in the pertinent documents, is delivered in a natural language format comprehensible to all stakeholders. These search results feed into the text understanding for the Large Language Model (LLM), enabling it to provide an accurate response to the user's query from the extensive log collection.

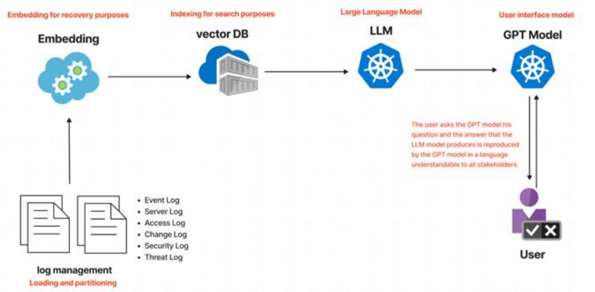

Figure 3. Report analysis using local LLMs

5. Assessment

To provide a comprehensive overview of the utility of this model, along with its notable advantages, which include exceptional text comprehension, proficiency in handling unstructured data and complex patterns, continuous learning capabilities, and the generation of natural language output for improved comprehension among non-technical stakeholders, three primary advantages emerge:

First: Enhanced Log Security

One of the key benefits lies in log security. This model is implemented locally, eliminating the need to transmit logs outside the organization or outsource their analysis. This inherent security feature ensures that sensitive log data remains within the organization's control.

Second: Time Efficiency

The model offers a substantial reduction in the time required to extract necessary information from logs. In traditional log analysis, composing several query lines demands time for contemplation and typing. However, with the assistance of the large language model, formulating a question in natural language requires significantly less time. To quantify this time-saving advantage, we considered 24 distinct non-repetitive questions to be written as queries. The time taken for each query was compared with the time it would take a person to naturally pose the same questions to the model. Initially, users working with the model may spend slightly more time formulating questions. However, over time, as users gain experience, Chart 1 depicts a noticeable trend toward reduced query time. This assessment underscores the model's capacity to enhance both log security and operational efficiency, making it a valuable asset for organizations seeking to streamline their log management and analysis processes.

Chart 1. Comparison of time consumption in each method

Third: Cost Reduction
The second advantage discussed relates to the reduction in costs. Traditionally, writing a suitable query for log analysis necessitates the presence of an expert with relevant knowledge within the organization, responsible for continuous analysis and necessary checks. However, with the language model, anyone can pose a question without the need for comprehensive expertise, and receive a suitable answer. This capability contributes to cost savings by obviating the need to hire human resources for log analysis.
In Chart 2, presented as a cumulative frequency chart, a cost comparison is depicted. The chart illustrates the requirements, implementation, and maintenance costs over two years for both the traditional model and the language model in a private company. Initially, the language model incurs higher implementation costs, but as time progresses, only maintenance costs are incurred, demonstrating a declining trend. Conversely, in the traditional model, the cost over two years, primarily related to hiring an expert, is nearly twice that of the language model. This third advantage highlights the substantial cost savings achievable by adopting the language model for log analysis, offering a more cost-effective and accessible approach to organizations.

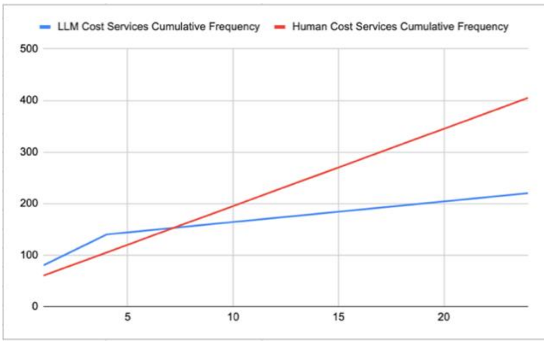

Chart 2. Cost comparison in each method in the form of cumulative frequency

6. Limitations

Despite significant progress and advancements in the field of log analysis with the aid of artificial intelligence, several challenges persist, offering opportunities for future research and the expansion of these methods. The following are notable challenges, along with potential solutions:
Generalization Ability: Achieving a valid output, in addition to log data, necessitates the inclusion of more information for model training. This expanded dataset increases the workload of the ensemble, thereby demanding additional research efforts.
Interpretability: Enhancing methods and models that make the behavior and predictions of machine learning systems comprehensible to humans is an ongoing research endeavor [27]. Improving interpretability is crucial in ensuring that AI-driven log analysis can be effectively understood and utilized.
Resource Consumption: GAI models face significant challenges related to resource consumption, including the computational power required for model training and operational costs, primarily concerning energy consumption during model operation [28]. Research efforts should focus on cost-effective learning methods to mitigate these resource-related challenges.
Security: Security remains a paramount concern in log management and analysis across all organizational departments. Entrusting log data to external entities or utilizing non-local tools increases security risks. GAI security models often exhibit "black box" features, raising security concerns [29]. To address these concerns and safeguard data privacy and intellectual property, organizations must implement security measures related to production models and strengthen defenses against potential attacks. This article

proposes the establishment of a local structure to address these security issues.

Limitations in Large Language Models (LLMs): LLMs sometimes produce unrealistic yet seemingly acceptable predictions, referred to as hallucinations [30]. To tackle this issue, new research directions, such as Augmented Language Models (ALMs), are explored [31]. ALMs are language models equipped with enhanced reasoning skills and the ability to utilize external tools and modules, expanding their processing capabilities and potentially mitigating hallucination-related limitations.

These challenges present opportunities for further research and innovation in the field of log analysis with the assistance of artificial intelligence, ultimately enhancing the capabilities and effectiveness of log analysis systems.

7. Conclusion

By harnessing the latest advancements in artificial intelligence technologies and locally deploying a large language model, without the need for outsourcing, organizations can securely engage with logs originating from various departments, such as network and data center management, marketing, human resources, finance, and accounting. These logs are systematically collected, stored, and efficiently managed, making them readily available for analysis through the implementation of a large language model. This approach culminates in the establishment of an accurate and effective system for addressing inquiries based on logs. It addresses the limitations of traditional log analysis by significantly reducing the time required for analysis. Moreover, it allows users to obtain answers simply by framing questions in natural language. This streamlined approach also leads to substantial cost savings by eliminating the necessity of hiring specialized experts. Taken together, these measures mark a significant stride toward organizational growth and success, enhancing the efficiency of log analysis and information retrieval, ultimately contributing to improved decision-making and operational effectiveness.

References

[1] M. Landauer, S. Onder, F. Skopik, and M. Wurzenberger, "Deep learning for anomaly detection in log data: A survey," *Machine Learning with Applications*, vol. 12, p. 100470, Jun. 2023, doi: 10.1016/J.MLWA.2023.100470.

[2] S. Muthurajkumar, S. Ganapathy, M. Vijayalakshmi, and A. Kannan, "Secured Temporal Log Management Techniques for Cloud," *Procedia Comput Sci*, vol. 46, pp. 589–595, Jan. 2015, doi: 10.1016/J.PROCS.2015.02.098.

[3] R. Meyers, "Data highway and the digital transformation: arguments for secure, centralised log management," *Network Security*, vol. 2020, no. 10, pp. 17–19, Oct. 2020, doi: 10.1016/S1353-4858(20)30119-7.

[4] Q. Wang, X. Zhang, X. Wang, and Z. Cao, "Log Sequence Anomaly Detection Method Based on Contrastive Adversarial Training and Dual Feature Extraction," *Entropy*, vol. 24, no. 1, 2022, doi: 10.3390/e24010069.

[5] H. J. Liao, C. H. Richard Lin, Y. C. Lin, and K. Y. Tung, "Intrusion detection system: A comprehensive review," *Journal of Network and Computer Applications*, vol. 36, no. 1, pp. 16–24, Jan. 2013, doi: 10.1016/J.JNCA.2012.09.004.

[6] T. Niesen, S. Dadashnia, P. Fettke, and P. Loos, "A vector space approach to process model matching using insights from natural language processing," *Multikonferenz Wirtschaftsinformatik (MKWI)*, pp. 93–104, 2016.

[7] N. Wang, S. Sun, and D. OuYang, "Business Process Modeling Abstraction Based on Semi-Supervised Clustering Analysis," *Business & Information Systems Engineering*, vol. 60, Jul. 2018, doi: 10.1007/s12599-016-0457-x.

[8] M. Landauer, F. Skopik, M. Wurzenberger, and A. Rauber, "System log clustering approaches for cyber security applications: A survey," *Comput Secur*, vol. 92, p. 101739, May 2020, doi: 10.1016/J.COSE.2020.101739.

[9] M. Landauer, M. Wurzenberger, F. Skopik, G. Settanni, and P. Filzmoser, "Dynamic log file analysis: An unsupervised cluster evolution approach for anomaly detection," *Comput Secur*, vol. 79, pp. 94–116, Nov. 2018, doi: 10.1016/J.COSE.2018.08.009.

[10] M. and M. Q. and H. B. and L. T. Dai Fei and Liu, "Refactor Business Process Models for Efficiency Improvement," in *Cloud Computing, Smart Grid and Innovative Frontiers in Telecommunications*, G. and Q. M. and X. W. and H. T. Zhang Xuyun and Liu, Ed., Cham: Springer International Publishing, 2020, pp. 454–467.

[11] W. Xu, L. Huang, A. Fox, D. Patterson, and M. I. Jordan, "Detecting Large-Scale System Problems by Mining Console Logs," in *Proceedings of the ACM SIGOPS 22nd Symposium on Operating Systems Principles*, in SOSP '09. New York, NY, USA: Association for Computing Machinery, 2009, pp. 117–132. doi: 10.1145/1629575.1629587.

[12] Q. Lin, H. Zhang, J.-G. Lou, Y. Zhang, and X. Chen, "Log Clustering Based Problem Identification for Online Service Systems," in *Proceedings of the 38th International Conference on Software Engineering Companion*, in ICSE '16. New York, NY, USA: Association for

Computing Machinery, 2016, pp. 102–111. doi: 10.1145/2889160.2889232.

[13] J.-G. Lou, Q. Fu, S. Yang, Y. Xu, and J. Li, "Mining invariants from console logs for system problem detection," in *2010 USENIX Annual Technical Conference (USENIX ATC 10)*, 2010.

[14] X. Zhang *et al.*, "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*, in ESEC/FSE 2019. New York, NY, USA: Association for Computing Machinery, 2019, pp. 807–817. doi: 10.1145/3338906.3338931.

[15] M. Du, F. Li, G. Zheng, and 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*, in CCS '17. New York, NY, USA: Association for Computing Machinery, 2017, pp. 1285–1298. doi: 10.1145/3133956.3134015.

[16] W. Meng *et al.*, "Loganomaly: Unsupervised detection of sequential and quantitative anomalies in unstructured logs.," in *IJCAI*, 2019, pp. 4739–4745.

[17] H. Gimpel *et al.*, "Unlocking the power of generative AI models and systems such as GPT-4 and ChatGPT for higher education: A guide for students and lecturers," Universität Hohenheim, Fakultät Wirtschafts- und Sozialwissenschaften, Stuttgart, 2023.s

[18] Y. Liu *et al.*, "Generative artificial intelligence and its applications in materials science: Current situation and future perspectives," *Journal of Materiomics*, vol. 9, no. 4, pp. 798–816, Jul. 2023, doi: 10.1016/J.JMAT.2023.05.001.

[19] A. Radford *et al.*, "Language models are unsupervised multitask learners," *OpenAI blog*, vol. 1, no. 8, p. 9, 2019.

[20] T. Brown *et al.*, "Language Models are Few-Shot Learners," in *Advances in Neural Information Processing Systems*, H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin, Eds., Curran Associates, Inc., 2020, pp. 1877–1901.

[21] L. Ouyang *et al.*, "Training language models to follow instructions with human feedback," in *Advances in Neural Information Processing Systems*, S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh, Eds., Curran Associates, Inc., 2022, pp. 27730–27744.

[22] A. Gilson *et al.*, "How Does ChatGPT Perform on the Medical Licensing Exams? The Implications of Large Language Models for Medical Education and Knowledge Assessment," *medRxiv*, 2022, doi: 10.1101/2022.12.23.22283901.

[23] Y. K. Dwivedi *et al.*, "Opinion Paper: 'So what if ChatGPT wrote it?' Multidisciplinary perspectives on opportunities, challenges and implications of generative conversational AI for research, practice and policy," *Int J Inf Manage*, vol. 71, p. 102642, Aug. 2023, doi: 10.1016/J.IJINFOMGT.2023.102642

[24] E. A. M. Van Dis, J. Bollen, W. Zuidema, R. van Rooij, and C. L. Bockting, "ChatGPT: five priorities for research," *Nature*, vol. 614, no. 7947, pp. 224–226, 2023.

[25] C. Qin, A. Zhang, Z. Zhang, J. Chen, M. Yasunaga, and D. Yang, "Is ChatGPT a general-purpose natural language processing task solver?," *arXiv preprint arXiv:2302.06476*, 2023.

[26] S. Bubeck *et al.*, "Sparks of artificial general intelligence: Early experiments with gpt-4," *arXiv preprint arXiv:2303.12712*, 2023.

[27] Y. Liu *et al.*, "Cloud-VAE: Variational autoencoder with concepts embedded," *Pattern Recognit*, vol. 140, p. 109530, Aug. 2023, doi: 10.1016/J.PATCOG.2023.109530.

[28] A. Brock, J. Donahue, and K. Simonyan, "Large scale GAN training for high fidelity natural image synthesis," *arXiv preprint arXiv:1809.11096*, 2018.

[29] D. Ganguli *et al.*, "Predictability and Surprise in Large Generative Models," in *Proceedings of the 2022 ACM Conference on Fairness, Accountability, and Transparency*, in FAccT '22. New York, NY, USA: Association for Computing Machinery, 2022, pp. 1747–1764. doi: 10.1145/3531146.3533229.

[30] S. Welleck, I. Kulikov, S. Roller, E. Dinan, K. Cho, and J. Weston, "Neural text generation with unlikelihood training," *arXiv preprint arXiv:1908.04319*, 2019.

[31] G. Mialon *et al.*, "Augmented language models: a survey," *arXiv preprint arXiv:2302.07842*, 2023.

[32] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," *arXiv preprint arXiv:1810.04805*, 2018.

[33] A. Chowdhery *et al.*, "Palm: Scaling language modeling with pathways," *arXiv preprint arXiv:2204.02311*, 2022.

[34] M. Chen *et al.*, "Evaluating large language models trained on code," *arXiv preprint arXiv:2107.03374*, 2021.