Proceedings of the 6th International Conference on Mobile Computing and Sustainable Informatics (ICMCSI-2025)
IEEE Xplore Part Number: CFP25US4-ART; ISBN: 979-8-3315-2266-7

Automated AI Tool for Log File Analysis

Prerna Lohar
dept.of Computer Engineering & Technology
Dr. Vishwanath Karad, MIT World Peace University
Pune, India
prerna.lohar@mitwpu.edu.in

Trupti Baraskar
dept.of Computer Engineering & Technology
Dr. Vishwanath Karad, MIT World Peace University
Pune, India
trupti.baraskar@mitwpu.edu.in

2025 6th International Conference on Mobile Computing and Sustainable Informatics (ICMCSI) | 979-8-3315-2266-7/25/$31.00 ©2025 IEEE | DOI: 10.1109/ICMCSI64620.2025.10883511

***Abstract*— Log file analysis has a critical role in monitoring and maintaining software systems, yet the manual inspection of logs becomes increasingly impractical with the growing volume of data. This survey paper explores recent advancements in automated log file analysis, with a particular focus on the integration of AI techniques, which includes ML models and NLP. The study identifies key challenges in traditional methods, such as the need for human interpretation, difficulty in detecting new errors, and issues in backtracking within continuous data streams. Moreover, we examine state-of-the-art AI approaches, like LLaMA 2, to streamline log analysis by automating error detection, summarization, and anomaly identification. Research deficiencies are identified, notably the necessity for advanced methodologies to manage variety of log formats, automated model optimization, and ongoing learning processes. This comprehensive review endeavors to provide a thorough examination of the current landscape, encompassing perspectives on potential outcomes and prospective trajectories in the domain of AI-enhanced log file analysis. In addition to providing insights into prospective solutions and future directions in AI-driven log file analysis, this study attempts to give an in-depth analysis of the current situation.**

Keywords— LLM, Log Analysis, Error Detection, Log Parsing

I. INTRODUCTION

A. Analysis of Artificial Log Files for System Security and Effectiveness

Artificial log file analysis is now important in preserving system security, dependability, and operational effectiveness due to the growing size and complexity of contemporary software systems. Traditional log analysis methods often rely on heuristic-based parsers, which are limited by manual configuration requirements and inflexible rules that struggle with the diverse formats of real-world log data [1, 12].

B. Large Language Model (LLM) Framework Developments for Improved Log Analysis

In an effort to overcome these constraints, recent developments like LogBatcher use Large Language Models (LLMs) to increase accuracy and reduce the requirement for preset demonstrations. Even while LLM-based systems have a lot of promise, issues with scalability, contextual awareness, and adaptability to new log formats still exist [2]. LLMs have demonstrated transformative power, particularly in cybersecurity and anomaly detection, by providing interpretability, adaptability, and advanced feature extraction capabilities [2,8]. For example, CYGENT—a conversational agent for log analysis—utilizes GPT-3.5 to convert complex log data into understandable outputs for security analysts, though scalability and adaptability to new log structures still require further development [4]. Frameworks like NeuralLog further innovate by bypassing traditional parsing methods, and directly processing raw log data using Transformer models, which improves the handling of out-of-vocabulary terms and context capture [8].

C. Filling Important Gaps in Error Detection and Real-Time Log Backtracking

Despite advancements in LLM-based log analysis, limitations remain in real-time log backtracking, managing heterogeneous log sources, and addressing the scarcity of labeled datasets for training robust anomaly detection models [3, 6, 10]. The AI-driven tool that uses LLaMA 2 for direct log reading, backtracking, and analysis, this study aims to close this difference and enable error identification without the need for pretrained parsing rules. This tool seeks to increase the efficiency and sustainability of automated log file analysis by automating error discovery, enhancing backtracking accuracy, and dynamically adapting to changing system behaviors through the integration of sophisticated NLP and machine learning approaches.

D. Significance of LLM in addressing the log files

A key role in the suggested tool for automating log file analysis is performed by artificial intelligence (AI). AI helps the system's ability to preprocess raw logs, find errors, and detect abnormalities with great accuracy by using sophisticated models like LLaMA 2. Through a process of iterative fine-tuning and enhanced contextual comprehension, AI transfigures the traditionally manual and heuristic-oriented methodology of log analysis into a streamlined, automated workflow.

II. LITERATURE REVIEW

[2], In order to enhance the automated analysis of logs produced by large-scale software systems, the research document "Stronger, Cheaper: LLM-Based Parsing" presents LogBatcher, a unique log parsing framework. While modern techniques employing large language models (LLMs) rely largely on demonstration cases, which results in inefficiencies, traditional log parsers frequently rely on

heuristics and require substantial tuning. By processing batches of log messages using the latent properties of log data, LogBatcher overcomes these problems by being training- and demonstration-free and improving the LLM's comprehension of the context and linkages in the logs. Multiple tests on several public log datasets demonstrate that LogBatcher surpasses existing progressive approaches in terms of accuracy and cost-effectiveness, making it a viable choice for real-world log parsing applications.

[1], The use of LLMs for log analysis in cybersecurity is examined in this study, which provides a thorough benchmark of five LLMs on six datasets. The study shows that these models perform much better in sequence classification tasks when they are fine-tuned, with DistilRoBERTa obtaining the best accuracy. The study demonstrates the benefits of LLMs over conventional log analysis techniques by utilizing the LLM4Sec pipeline, especially with regard to interpretability and flexibility in handling different log formats. The results highlight how LLMs might enhance security monitoring and anomaly detection while also pointing out shortcomings in existing approaches that need more research.

[3], With the use of cutting-edge GPT-3.5 turbo models, this study presents CYGENT, a smart conversational agent intended to support cybersecurity security experts. The framework's goal is to automate log file analysis and summarization by utilizing Large Language Models to convert complex data into formats that are easy for humans to understand. In addition to pointing out shortcomings in the existing application of LLMs for log analysis and conversational agents, the paper emphasizes how well this method works to increase understanding and productivity in cybersecurity operations. The study illustrates the potential of this novel methodology to improve cybersecurity professionals' capacity to handle the growing complexity of IT and IoT settings using stringent evaluation metrics.

[6], To increase the performance and effectiveness of detecting system problems, this study presents an LSTM-based model for anomaly identification in log analysis. It employs methods like LogWord2Vec and data augmentation to address the critical issue of synonym identification in log entries. The results includes an automatic log classification which is more effective for system monitoring and troubleshooting.

[4] Pan J. investigates the use of ML methods to log file analysis in order to identify irregularities and comprehend user activity. It examines a number of approaches, points out drawbacks, including the difficulty in choosing a threshold, and indicates important research gaps, especially in the areas of log maintenance and dataset quality. The results are intended to increase log analysis's efficacy in enhancing operational effectiveness and system security. In order to avoid the requirement for log parsing, which is frequently prone to errors because of out-of-vocabulary words and semantic misunderstandings, this research article introduces NeuralLog, a unique approach for log-based anomaly identification. NeuralLog uses a Transformer-based classification model and BERT for semantic representation to precisely identify abnormalities by using raw log messages. The empirical study demonstrates that NeuralLog outperforms existing methods for high F1-scores on multiple public datasets, filling a major anomaly detection gap and fixing basic errors in log analysis techniques.

[8] presents a machine learning toolbox that uses unsupervised learning methods like K-means clustering and truncated SVD to examine server and desktop log files with the purpose of identifying malicious activity. With a high accuracy rate of 98.56% in detecting malicious entries, the toolkit shows a notable decrease in log file sizes. The study highlights the necessity for creative machine learning methods in cybersecurity by addressing the difficulties that conventional Intrusion Detection Systems (IDS) experience in handling growing encrypted network traffic.

[9] introduced a Log-based Intrusion Detection System (LIDS) that uses network log analysis to forecast cyberattacks in order to improve information security. The authors draw attention to the difficulties that businesses encounter as a result of the growing complexity of cyber threats and suggest a system that uses ML methods like DNN and Decision Trees to aggregate data from several sources. The project intends to enhance the detection of abnormal traffic by concentrating on feature selection and utilizing the KDD Cup 1999 dataset. This will address present limits in intrusion detection approaches and highlight gaps for future research in responding to growing cyber threats.

[10] In order to facilitate analysis, this study focuses on log parsing approaches that convert raw, unstructured log data into organized representations. In order to extract patterns and events and provide focused insights inside particular log types, it mostly uses rule-based techniques. This method, however, is limited in its ability to handle various log formats and depends on static rules, which may not be able to handle data that changes dynamically. In order to improve the speed and precision of mistake detection across diverse systems, the study finds a need for more flexible log parsers that can handle several formats in real time.

[11] In order to comprehend and characterize user and system interactions, the 2020 research investigates behavioral profiling through log analysis. The research creates profiles for various system operations by using log pattern matching to identify recurrent behavioral patterns in system and application logs. This method lacks adaptability to new behaviors and log sources and is constrained by scaling concerns despite its efficacy in some situations. The study emphasizes how crucial it is to have scalable, flexible log analysis techniques that can keep up with the massive amounts of log data and changing system behaviors.

[12] Using a Transformer-based model that makes use of self-supervised learning and reinforcement learning strategies, this study offers a novel approach to log analysis. It overcomes the challenges of recognizing diverse log sources and the absence of labeled datasets by letting the model learn from regular log entries and updheaate itself with labeled instances as they become available. According to the results, the suggested approach can successfully improve anomaly detection skills in practical applications, strengthening system resilience against errors and cyberattacks.

III. RESEARCH GAP

A. *Lack of research in advanced LLM*

A review of previous studies revealed a number of significant shortcomings in the log analysis techniques now in use, which this study seeks to fill. Research indicates that there is little investigation of another large language model (LLM) architectures, with a significant emphasis on traditional models such as BERT and RoBERTa. This limited strategy raises the possibility of investigating other models, like LLaMA 2, which might have special advantages in the analysis of intricate log data. Furthermore, labeled data and demonstration examples are frequently the mainstays of traditional log parsers, which results in inefficiencies and higher operating expenses. In order to lessen the requirement for significant labeling and make the system both affordable and useful for real-world applications, this research aims to build a demonstration-free parsing approach.

B. *Conversational Log Analysis Underutilization of LLMs*

The underuse of LLMs in interactive log summary and analysis tools, especially in conversational interfaces, is another finding of the survey. In order to close this gap, the suggested system will incorporate sophisticated LLMs to facilitate conversational, user-friendly interactions that make it simple for users to retrieve event summaries and trace problems back. Moreover, it is observed that existing log analysis techniques are not flexible enough to identify new anomalies as they appear. In order to ensure that the model adapts dynamically to changing log patterns and increases the accuracy of anomaly identification, this project will integrate a continuous learning mechanism.

C. *The absence of a unified workflow and sophisticated summarization methods*

The absence of a general workflow and unified data structure complicates log analysis by allowing duplication of processing activities. In order to create a more unified and effective analytical system, it attempts to expedite data preprocessing and develop interoperability across various log formats through a standard pipeline. It is clear that log summarization methods are required, highlighting the importance of putting these strategies into practice in order to distill enormous log data into intelligible, useful insights. This research will immediately address these issues by creating a comprehensive and adaptable log analysis framework that satisfies the demands of modern digital systems..

D. *Limitations with Existing Log File Analysis Techniques*

Current log file analysis approaches face several obstacles, including handling log format diversity, relying on large labeled datasets, and high false positive and negative rates in anomaly identification. These systems frequently exhibit a deficiency in contextual comprehension, adaptability to the dynamic nature of log patterns, and proficient integration of advanced artificial intelligence techniques, including large language models (LLMs).

IV. RESEARCH CONTRIBUTION

The building of an Automated AI Tool for Log File Analysis that makes use of sophisticated LLMs—more especially, LLaMA 2—to expedite mistake detection, anomaly identification, and log backtracking in a variety of extensive log datasets is the main contribution of this research. This tool eliminates the requirement for manual settings and large amounts of labeled data by integrating a flexible, demonstration-free log parsing technique. Continuous learning features enable it to dynamically adjust to changing log patterns and deliver real-time, approachable insights via a conversational interface. By filling important gaps in current log analysis techniques, this study provides a scalable and effective approach to anomaly management and proactive system monitoring.

V. PROPOSED CONCEPTUAL FRAMEWORK

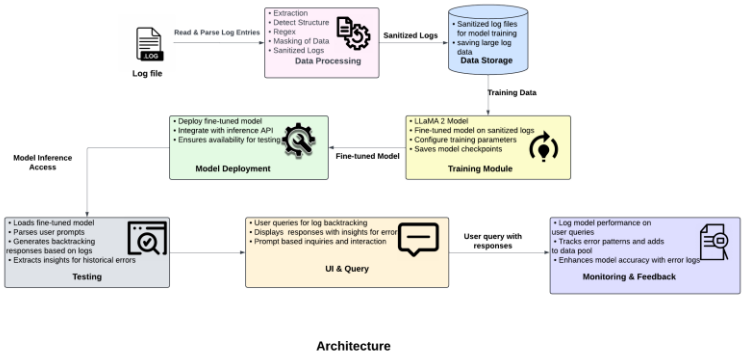

Fig no. 1.1 Architectural workflow of the tool

This conceptual architecture for an Automated AI-based Log File Analysis System in fig [1.1] provides a head-to-foot solution, from log ingestion to continuous refinement. The pipeline starts with Log File Ingestion, where raw logs are prepared for analysis through Data Processing, which involves extraction, structural detection, regex, and masking sensitive information, resulting in sanitized logs.

For additional analysis, logs are pre-processed to clean sensitive data and standardize various formats into a uniform schema.

Parameters like timestamps are used by the system to arrange logs chronologically, while error codes like 400 or 500 are used to identify abnormalities. Using factors like service error, it assesses the operational impact of problems and uses repair services to offer practical repair recommendations. Job processes, like job names and parent-

child relationships, are also gathered to provide more context. To ensure uniformity and enable clear and consistent processing, several log formats are standardized into a single JSON schema. Together, these characteristics reduce manual involvement and improve reliability by addressing the need for precise, automated, and context-aware log analysis.

These sanitized logs are stored in Data Storage as a repository for training data, crucial for handling large datasets in anomaly and pattern detection. In the Training Module, the LLaMA 2 model is fine-tuned on these logs, with checkpoints recorded for iterative improvements. The trained model then moves to Model Deployment, where it integrates with an inference API to allow real-time analysis. The LLaMA 2 model is refined throughout training to extract context and patterns from structured logs, allowing it to produce precise and contextually aware responses. In order to retrieve log attributes including error codes, task information, and system affects, the model handles user requests throughout the testing phase. Testing enables the model to process user prompts, generate backtracking responses, and provide insights into historical error patterns. A UI & Query Module allows user interaction with the model for targeted inquiries and displays model responses. The system also detects irregularities and traces faults to determine their underlying causes.

In order to increase accuracy by fine-tuning the LLaMA 2 model, the system uses LoRA to optimize resource consumption and discover patterns in log data. Logs that have been preprocessed and standardized guarantee reliable and consistent input for inference and training. The model's comprehension of mistakes and anomalies is further improved by iteratively fine-tuning over several epochs. During inference, structured prompts help the model concentrate on tasks, improving the accuracy of the responses.

AI is smoothly incorporated into the process at many points to improve the automation and effectiveness of log file analysis. AI analyzes the logs via preprocessing by standardizing various formats and hiding sensitive information, guaranteeing data privacy and consistency for jobs that come after. The LLaMA 2 model is optimized to identify patterns in log data during its training stage, enabling it to produce context-aware inference responses. Automation of the analysis of log data, finding errors, and detecting irregularities, the AI model reduces the need for manual intervention at the inference step. Plus, the AI system continuously learns and adjusts to new log formats and mistake patterns through repeated fine-tuning, assuring improved accuracy and dependability over time.

Additionally, the Monitoring & Feedback Module logs performance, tracks errors, and adjusts the model based on evolving log patterns to gradually increase accuracy. For automated log analysis, this framework integrates user interaction, AI model training, data processing, and adaptive feedback.

VI. DISCUSSION

The optimized LLaMA 2 model can produce actionable insights and detects abnormalities, including failures. The model exhibits enhanced accuracy and automation in log analysis by dynamically backtracing errors and correlating events, in contrast to conventional approaches that depend on static rules.

Aspects	Methods	
	Existing Models	***LLaMA 2 - Model***
Scalability	inefficient diverse datasets	handles large scale logs and adapts dynamically
Insights	lack of contextual understanding	provide detailed insights and backtracking outputs
Error detection	prone to errors, relies on predefined rules	Identifies anomalies using AI patterns from training (e.g http 400)
Preprocessing	manual masking and formatting of logs	Automated log sanitization and JSON conversion

Fig no 1.2 Traditional Methods vs Proposed System

VII. CONCLUSION AND FUTURE SCOPE

The Automated AI-based Log File Analysis System offers a complete answer to the problems of tracking errors, analyzing log data, and detecting anomalies. The integration of LLaMA 2 model with advanced data processing methods, the system significantly enhances the analysis and interpretation of log entries from many sources. The architecture enables a seamless transition to end-user involvement from log input to data processing, model training, and real-time deployment. Both preventative and reactive error detection are made possible by this integrated strategy, which gives consumers useful, actionable outcomes. The system's feedback mechanism can be use for development, enabling the model to upgrade and strengthen against intricate log patterns. Also, this framework contributes to handle, examine, and get value from massive amounts of log data.

By incorporating a format-agnostic preprocessing module to standardize logs into a common schema, the suggested research is sufficiently resilient to handle a variety of log formats in the future. In cross-domain generalization, the model will be trained on a variety of structured and unstructured datasets. While a feedback loop will continuously improve format handling for increased resilience and scalability, strategies like template matching, embedding-based log representation, and transfer learning will guarantee adaptability to unforeseen forms.

The Automated AI-based Log File Analysis System may be improved in the future to increase user accessibility, scalability, and adaptability. It would be able to handle a variety of log kinds and industries by including more machine learning models. Proactive problem-solving might be made possible by real-time anomaly detection, and scalability for bigger datasets could be improved by utilizing cloud or edge computing. Improvements in natural language processing (NLP) could make the system easier for non-technical users to use, and integration with cybersecurity technologies, including SIEM systems, could increase its usefulness for threat detection. Last but not least, a self-learning feedback loop would enable the model to develop on its own, eliminating the need for human

retraining and guaranteeing accuracy and robustness over time.

REFERENCES

[1] Karlsen, Egil, Xiao Luo, Nur Zincir-Heywood, and Malcolm Heywood. "Benchmarking Large Language Models for Log Analysis, Security, and Interpretation." Journal of Network and Systems Management 32, no. 3 (2024): 59.

[2] Xiao, Yi, Van-Hoang Le, and Hongyu Zhang. "Stronger, Faster, and Cheaper Log Parsing with LLMs." arXiv preprint arXiv:2406.06156 (2024).

[3] Balasubramanian, Prasasthy, Justin Seby, and Panos Kostakos. "CYGENT: A cybersecurity conversational agent with log summarization powered by GPT-3." arXiv preprint arXiv:2403.17160 (2024).

[4] Pan, Jonathan. "AI based Log Analyser: A Practical Approach." arXiv preprint arXiv:2203.10960 (2022).

[5] Cheng, Qian, Amrita Saha, Wenzhuo Yang, Chenghao Liu, Doyen Sahoo, and Steven Hoi. "Logai: A library for log analytics and intelligence." arXiv preprint arXiv:2301.13415 (2023).

[6] Ramachandran, Shekar, Rupali Agrahari, Priyanka Mudgal, Harshita Bhilwaria, Garth Long, and Arisha Kumar. "Automated log classification using deep learning." Procedia Computer Science 218 (2023): 1722-1732.

[7] Korzeniowski, Łukasz, and Krzysztof Goczyła. "Landscape of automated log analysis: A systematic literature review and mapping study." IEEE Access 10 (2022): 21892-21913.

[8] Abdalla, Rawand Raouf, and Alaa Khalil Jumaa. "Log File Analysis Based on Machine Learning: A Survey: Survey." UHD Journal of Science and Technology 6, no. 2 (2022): 77-84.

[9] Zhang, Tianzhu, Han Qiu, Gabriele Castellano, Myriana Rifai, Chung Shue Chen, and Fabio Pianese. "System log parsing: A survey." IEEE Transactions on Knowledge and Data Engineering 35, no. 8 (2023): 8596-8614.

[10] Zhao, Zhijun, Chen Xu, and Bo Li. "A LSTM-based anomaly detection model for log analysis." Journal of Signal Processing Systems 93, no. 7 (2021): 745-751.

[11] Meier, Heidi, Eno Tõnisson, Marina Lepp, and Piret Luik. "Behaviour patterns of learners while solving a programming task: An analysis of log files." In 2020 IEEE Global Engineering Education Conference (EDUCON), pp. 685-690. IEEE, 2020.

[12] Debnath, Biplob, Mohiuddin Solaimani, Muhammad Ali Gulzar Gulzar, Nipun Arora, Cristian Lumezanu, Jianwu Xu, Bo Zong, Hui Zhang, Guofei Jiang, and Latifur Khan. "LogLens: A real-time log analysis system." In 2018 IEEE 38th international conference on distributed computing systems (ICDCS), pp. 1052-1062. IEEE, 2018.

[13] Ritchey, Ralph P., and Richard Perry. "Machine learning toolkit for system log file reduction and detection of malicious behavior." In IEEE INFOCOM 2021-IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS), pp. 1-2. IEEE, 2021.

[14] Zhu, Jieming, Shilin He, Jinyang Liu, Pinjia He, Qi Xie, Zibin Zheng, and Michael R. Lyu. "Tools and benchmarks for automated log parsing." In 2019 IEEE/ACM 41st International Conference on Software Engineering: Software Engineering in Practice (ICSE-SEIP), pp. 121-130. IEEE, 2019.

[15] Wang, Mengying, Lele Xu, and Lili Guo. "Anomaly detection of system logs based on natural language processing and deep learning." In 2018 4th International Conference on Frontiers of Signal Processing (ICFSP), pp. 140-144. IEEE, 2018.