2025 25th International Conference on Software Quality, Reliability, and Security Companion (QRS-C) | 978-1-6654-7773-4/25/$31.00 ©2025 IEEE | DOI: 10.1109/QRS-C65679.2025.00103

A Novel Semi-Supervised Model for Generalizing Log Anomaly Detection with Limited Labeled Data

Yicheng Sun[1], Jacky Keung[1], Zhen Yang[2,*], Shuo Liu[1], and Wing Kwong Chan[1]
[1]Department of Computer Science, City University of Hong Kong, Hong Kong, China
[2]School of Computer Science and Technology, Shandong University, Qingdao, China
{yicsun2-c, sliu273-c}@my.cityu.edu.hk, {jacky.keung, wkchan}@cityu.edu.hk, zhenyang@sdu.edu.cn
*corresponding author

Abstract—Log anomaly detection is critical for ensuring the reliability and stability of large-scale distributed systems by identifying unusual patterns in system logs. However, existing methods often fail to generalize across diverse datasets, hindering their scalability and robustness. This paper proposes a novel anomaly detection model that integrates an LLM-based log parser, Generative Adversarial Networks (GANs) for anomaly detection, and Permutation Event Modeling (PEM) to capture temporal dependencies in log events, using only 5% of labeled logs for training. Our model leverages GANs to generate synthetic log data, enabling it to generalize effectively across diverse datasets with varying system architectures and log formats. By reducing the reliance on labeled data and enhancing generalization, our approach improves performance and scalability, offering a robust solution for log anomaly detection in dynamic environments.

Keywords–Log anomaly detection; Log parsing; Semi-supervised learning; Generative adversarial networks (GANs)

1. Introduction

Accurate log anomaly detection is crucial for maintaining software reliability, as undetected anomalies can lead to system downtime, security breaches, or performance degradation [1]. Logs generated by various system components contain valuable insights into system behavior, errors, and performance. However, a significant challenge in this domain is the poor generalization of existing models across diverse log datasets. For example, models that perform well on public datasets like LogHub [2], particularly on HDFS, often fail to generalize effectively to other datasets such as BGL. This disparity arises from inherent differences in system architecture, log formats, and operational conditions [1]. Generative Adversarial Networks (GANs) [3] are particularly advantageous for generalizing across diverse log datasets because they can generate synthetic log data that mimics normal system behavior, enabling the model to detect anomalies as deviations from this generated distribution. Unlike other models, which often struggle with high variability in log formats and system behaviors [4], GANs do not rely on predefined rules or fixed patterns, making them more adaptable to the inherent diversity of log data. However, despite these advantages, the specific performance of GANs in log anomaly detection, especially across varying system architectures, log formats, and operational conditions, has not been thoroughly explored in existing research.

To address these challenges, we propose the use of GANs for anomaly detection, which is well-suited for modeling the distribution of normal log events. In GANs, the generator learns to produce synthetic log sequences that resemble normal data, while the discriminator detects deviations from this normal behavior, identifying anomalies. This approach is particularly effective because GANs can learn from very limited data, making them a powerful tool in scenarios where labeled data is scarce. However, applying GANs to log anomaly detection presents several challenges, including the need for high-quality, contextually rich log data to ensure accurate model training. To address this, we use an LLM-based parser for log parsing, ensuring high accuracy in event extraction. Additionally, GANs face difficulties in capturing the sequential and temporal dependencies inherent in log data, which are crucial for detecting complex anomalies. To overcome this, we employ 5% of labeled logs to build a diversified training set, reducing the inefficiency of manual labeling, and introduce Permutation Event Modeling (PEM), a method that captures event order and dependencies. PEM expands the training sample and enhances the model's ability to detect subtle anomalies, thereby improving the GAN's performance in sequential log data analysis. Overall, our primary contributions can be summarized in these key points:

- We propose a novel semi-supervised anomaly detection model that addresses the challenge of limited labeled log data by utilizing only 5% of labeled logs for training.
- We introduce a GAN-based approach with PEM, enabling the model to address poor generalization across datasets and enhance anomaly detection.

2. Study Design

In this study, we propose a semi-supervised anomaly detection model that combines the power of ChatGPT for log parsing with a Generative Adversarial Network (GAN) framework and Permutation Event Modeling (PEM) for enhanced detection performance. As shown in Figure 1, the model is designed to address the challenges of limited labeled log data and poor generalization across different log datasets.

2693-9371/25/$31.00 ©2025 IEEE
DOI 10.1109/QRS-C65679.2025.00103

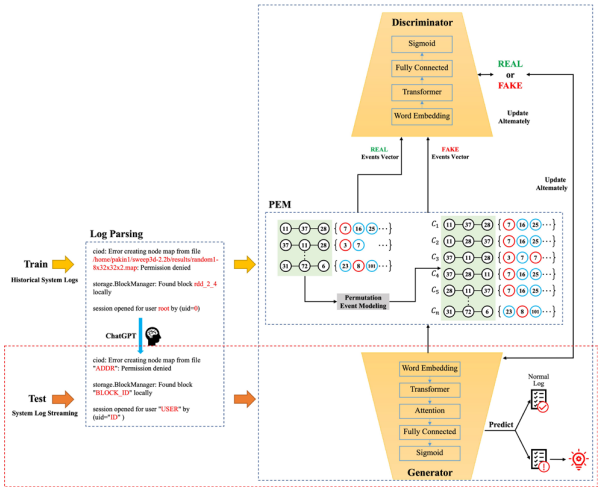

Figure 1. An overview of our proposed model.

2.1 Log Parsing Using ChatGPT

Previous studies have shown that the accuracy of log parsing significantly influences the performance of anomaly detection systems [5]. Rule-based methods, which rely on predefined patterns and conditions, have traditionally been used for log parsing, but they often struggle with the variability and complexity of modern log formats. In contrast, Large Language Models (LLMs), such as ChatGPT, have been shown to outperform rule-based methods in terms of parsing accuracy due to their ability to understand the context of log entries and extract meaningful event vectors more effectively [6]. Leveraging this advantage, we employ ChatGPT for log parsing, as its advanced capabilities enable it to capture the intricacies of log data more effectively.

2.2 Permutation Event Modeling (PEM)

PEM is employed to capture the temporal relationships and dependencies between log events. Unlike traditional methods that treat log events independently, PEM models the events as a sequence, taking into account the order and relationship between events in the log. This is crucial because anomalies often arise from unusual sequences or patterns of events that deviate from the expected system behavior. By permuting the event sequences, PEM can generate multiple permutations of the logs, helping to identify subtle deviations that may not be apparent when considering individual events. This approach enhances the model's ability to generalize across different log datasets, as it focuses on capturing the inherent structure within the logs rather than relying on fixed event patterns.

2.3 GAN Framework for Anomaly Detection

We incorporate a Generative Adversarial Network (GAN) [3] architecture into our model for anomaly detection. The GAN consists of two primary components: the generator and the discriminator. The generator aims to generate synthetic log sequences that resemble the normal operating behavior of the system. It takes as input the event vectors parsed by ChatGPT and processed by PEM, generating log sequences that follow the expected distribution of events. The discriminator, on the other hand, evaluates whether a given log sequence is real (from the actual logs) or fake (generated by the model). Using a sigmoid function in the final layer, the discriminator classifies the event sequences as either "real" or "fake," guiding the generator to improve its synthetic log sequences over time. The generator and discriminator are trained alternately, with the generator improving its ability to create realistic log sequences, while the discriminator becomes increasingly adept at distinguishing anomalies. The discriminator is fully connected and enhanced with transformer layers to capture the sequential dependencies within the logs, as well as word embedding layers to further process the log events. This adversarial process ensures that the generator learns the underlying distribution of log events with minimal labeled data, which is particularly useful for generalizing across different datasets and making the model suitable for deployment in diverse systems.

3. Conclusion

We propose a novel anomaly detection model that combines PEM for capturing event dependencies and a GAN framework for effective anomaly detection. To evaluate our model, we are conducting experiments comparing it with advanced anomaly detection models across LogHub datasets. Our experiments aim to explore the model's generalization capabilities across various environments and its robustness in processing log data with varying structures. Additionally, we use only 5% of labeled logs to build a diversified training set, unlike other semi-supervised models that typically require larger labeled datasets. We are currently analyzing the results to assess how our model improves performance and addresses the challenges of generalization across diverse log datasets.

References

[1] Y. Sun, J. W. Keung, Z. Yang, et al., "SemiRALD: A semi-supervised hybrid language model for robust anomalous log detection," Information and Software Technology, vol. 2025, art. no. 107743.

[2] J. Zhu, S. He, P. He, J. Liu, and M. R. Lyu, "Loghub: A large collection of system log datasets for AI-driven log analytics," Proc. IEEE 34th Int. Symp. Softw. Reliab. Eng. (ISSRE), pp. 355–366, 2023.

[3] I. Goodfellow, J. Pouget-Abadie, M. Mirza, et al., "Generative adversarial networks," Commun. ACM, vol. 63, no. 11, pp. 139–144, 2020.

[4] V.-H. Le and H. Zhang, "Log-based anomaly detection with deep learning: How far are we?," Proc. 44th Int. Conf. Softw. Eng., pp. 1356–1367, 2022.

[5] Z. A. Khan, D. Shin, D. Bianculli, and L. Briand, "Guidelines for assessing the accuracy of log message template identification techniques," Proc. 44th Int. Conf. Softw. Eng., pp. 1095–1106, 2022.

[6] Z. Jiang, J. Liu, Z. Chen, Y. Li, J. Huang, Y. Huo, P. He, J. Gu, and M. R. Lyu, "Lilac: Log parsing using LLMs with adaptive parsing cache," Proc. ACM Softw. Eng., vol. 1, no. FSE, pp. 137–160, 2024.