Anomaly Detection in Log Files Using LLMs

Giorgio Saldana
Reykjavik University / Menntavegur 1, 102 Reykjavík, Iceland
giorgio24@ru.is

Abstract

The intersection of cybersecurity and Artificial Intelligence (AI) is increasingly significant in addressing modern challenges. In recent years, substantial efforts have been directed towards applying AI techniques such as Machine Learning, Deep Learning, and Natural Language Processing (NLP) to automate essential tasks in cybersecurity. However, traditional approaches often fall short in identifying complex patterns hidden within log files.

This paper explores the use of AI-driven techniques for log analysis to address existing challenges in the field. Using data from LogPai, log files are preprocessed and analyzed with advanced Transformer models, including BERT and LLAMA. The performance of these models is evaluated based on metrics such as accuracy, F1-score, precision, recall, and confusion matrices. The study highlights the effectiveness of open-source models as practical tools for log analysis, providing an alternative to proprietary systems. This work represents the culmination of an individual project focusing on the application of NLP in cybersecurity.

1 Introduction

The integration of Artificial Intelligence (AI) into cybersecurity is becoming increasingly critical, especially as the reliance on digital systems grows and the need to protect them from threats intensifies. Log anomaly detection plays a key role in cybersecurity by identifying unusual patterns in system log files that could indicate potential security breaches or operational failures. Traditional methods for analyzing logs, often rule-based or heuristic-driven, struggle to keep up with the complexity and sheer volume of modern log data. This project aims to address these challenges by applying advanced AI techniques—specifically, Large Language Models (LLMs)—to automate and enhance the effectiveness of log anomaly detection (Bommasani et al., 2021).

The application of techniques like Machine Learning (ML), Deep Learning (DL), and Natural Language Processing (NLP) to log analysis has shown considerable promise. However, there remains a need to better understand how more recent LLMs, such as BERT and LLAMA 3.2 1B, can effectively process and interpret the intricate sequences found in log data (Devlin et al., 2018) (Touvron et al., 2023). This project focuses on evaluating these models in classifying log traces as either anomalous or normal, providing insights into their strengths, nuances in their performance, and specific areas where each model performs best.

The objectives are threefold: (1) to demonstrate the potential of open-source LLMs for log anomaly detection, (2) to evaluate the robustness of these models using metrics such as precision, recall, and F1 score, and (3) to identify opportunities for further improvement in anomaly detection systems, including the incorporation of real-time analysis capabilities. This work contributes to the field of cybersecurity by showcasing how modern NLP techniques can enhance log analysis, with a strong emphasis on transparency, reproducibility, and performance.

The significance of this project lies in its ability to shift log anomaly detection from traditional methods to AI-driven solutions that can autonomously learn and adapt to complex log sequences. By leveraging open-source models such as LLAMA 3.2 1B and BERT, the findings are accessible and practical, enabling others to implement effective anomaly detection systems without relying on proprietary technologies. The methodology outlines the experimental setup, including dataset preparation, model selection, and evaluation strategies. Results focus on comparing BERT and LLAMA, highlighting LLAMA's slight advantage in performance. The discussion addresses these findings in depth, considers challenges like dataset imbalance, and suggests directions for im-

provement, such as enhanced generalization techniques and real-time applications. The conclusion underscores the value of LLMs in log anomaly detection and emphasizes the ongoing need for innovation in this area to strengthen cybersecurity defenses.

2 Background

This section introduces key concepts and terminology relevant to log anomaly detection, followed by a discussion of the scientific challenges inherent to this field of research.

2.1 Log Anomaly Detection

Log anomaly detection is a critical task within cybersecurity, focusing on identifying unusual patterns and behaviors in raw log data. Traditionally, rule-based models and heuristic approaches have been employed to detect patterns within log files. However, with the increasing complexity of data, more advanced methods involving Machine Learning (ML), Deep Learning (DL), and, more recently, Transformer-based approaches, have been explored. These advanced models are capable of capturing intricate patterns in high-dimensional spaces. Given that log files are text-based, Natural Language Processing (NLP) techniques, particularly Transformer models, have demonstrated superior performance across various evaluation metrics when applied to log anomaly detection tasks.

2.2 Large Language Models and Transformers

Large Language Models (LLMs) and Transformers have revolutionized the field of Natural Language Processing (NLP) by enabling machines to effectively process and understand textual data. Transformers, introduced through models like BERT and GPT, are designed to capture long-range dependencies and contextual relationships within sequences of text. They achieve this through mechanisms like self-attention, which allows the model to weigh the importance of different words in a sequence relative to each other.

LLMs, such as BERT, GPT, and newer models like LLAMA, are pre-trained on massive text corpora and can be fine-tuned for specific tasks, including log anomaly detection. Their ability to understand complex sequential data makes them particularly well-suited for analyzing log files, where patterns often span multiple entries and depend on contextual understanding. Unlike traditional approaches, these models can generalize across diverse datasets, making them adaptable to various logging formats and scenarios.

Transformers and LLMs also bring scalability and flexibility to cybersecurity tasks. By leveraging open-source models, practitioners can fine-tune these frameworks for specific challenges, such as detecting anomalous patterns in logs, without relying on proprietary systems. This adaptability ensures that solutions remain transparent and can be tailored to the unique needs of the domain.

2.3 Open Models

Open-source foundation models and Large Language Models (LLMs) have gained considerable attention in cybersecurity, particularly for tasks involving critical and sensitive data. Open models, such as LLaMA 3.2 and Mystral, provide publicly accessible alternatives to proprietary solutions like ChatGPT or Claude. Unlike closed models, where data is sent to external entities, open models offer enhanced transparency and control over data handling. With appropriate fine-tuning, these models can be adapted to address complex cybersecurity challenges, making them a viable option for tasks like log anomaly detection.

3 Methodology

The methodology outlines the steps taken in this project, from data collection to model training, evaluation, and the experimental setup utilizing high-performance computing resources. This section provides a detailed description of each component and the overall workflow. Figure 1 summarizes the process, highlighting the flow from data preparation to model training and evaluation for both BERT and LLAMA.

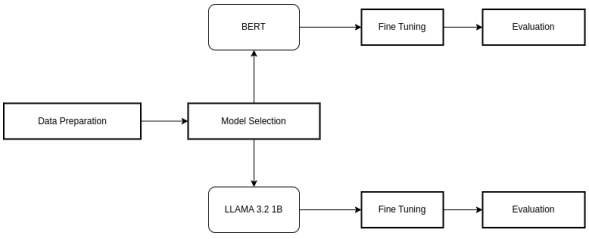

Figure 1: Flowchart of the Methodology for Log Anomaly Detection

3.1 Experimental Setup

The experiments were conducted using Icelandic HPC Elja, leveraging a high-performance environment specifically configured for deep learning experiments. The hardware used consisted of 1x NVIDIA A100 GPU with 40GB of VRAM, which is highly suited for the computational demands of training large language models (LLMs). Python served as the core programming language, along with several important libraries: PyTorch for model implementation, Transformers for handling pre-trained language models, Scikit-learn for evaluation metrics, Numpy for numerical computations, and Matplotlib with Seaborn for data visualization.

To manage resource allocation within the HPC environment, SLURM workload manager was employed, along with Lmod, a Lua-based module system, to handle environment configurations efficiently (Yoo et al., 2003). The scripts required to run the experiments were managed via bash scripts using the sbatch command for resource allocation and execution.

3.2 Dataset

The dataset used for this project is an open-source collection available on GitHub. It was originally sourced from a Hadoop Distributed File System (HDFS) and consists of approximately 2GB of labeled log traces (Shvachko et al., 2010). Each trace is categorized as either an anomaly or normal behavior, with rows containing sequences of features labeled from E1 to E29, along with their classification. The dataset includes several pre-existing auxiliary files to facilitate analysis:

- anomaly_label.csv: Contains trace labels, indicating whether each trace is normal or anomalous.
- Event_traces.csv: Provides detailed sequences of events for each trace.
- Event_occurrence_matrix.csv: Summarizes statistical distributions of events within the dataset.

For convenience and computational efficiency, the dataset was consolidated into a single .npz file for use during model training and evaluation.

3.3 Data Preparation

To prepare the dataset for model training, a series of preprocessing steps were applied. For BERT, the dataset was tokenized using the BERT tokenizer, which splits each log trace into meaningful tokens for input into the model. Additional feature engineering techniques were employed to ensure the models could effectively utilize the contextual information present in the log traces.

For LLAMA, the dataset was converted into a JSONL (JSON Lines) format, allowing the prompt-completion task to be structured appropriately. Each log trace was reformatted as a prompt with its expected continuation, ensuring the LLAMA model could perform classification tasks through its completion mechanism.

3.4 Model Selection

Two LLMs were chosen for this project: BERT and LLAMA 3.2 1B.

- **BERT**: BERT was selected for its strong contextual language understanding capabilities, making it suitable for tasks involving sequential and textual log data. The model was fine-tuned on the labeled log traces to classify them as anomalous or normal.
- **LLAMA 3.2 1B**: LLAMA, a GPT-based model, was evaluated for its effectiveness in handling prompt-based completion tasks. It was tasked with predicting the continuation of log traces and classifying them based on the structure and context of the logs.

Model training was conducted using PyTorch, with SLURM managing the computational resources on the Icelandic HPC. Hyperparameters, including the learning rate, batch size, and number of epochs, were optimized for each model to ensure strong performance.

3.5 Evaluation

The evaluation of the models was performed using several metrics to assess the effectiveness of each model in identifying anomalies in the log traces. Precision, recall, and F1 score were used as the primary metrics, providing insight into the accuracy, completeness, and overall performance of each model. In addition, a confusion matrix was plotted to visualize the model performance in terms of true positives, false positives, true negatives, and false negatives.

3.6 Reproducibility

The project is fully reproducible, with all necessary scripts and instructions made publicly

available at https://github.com/giorgiosld/Log-Anomaly-Detection-via-LLMs. The repository includes scripts for data preprocessing, model training, and evaluation, along with bash scripts for execution within an HPC environment. Public datasets, open-source models, and well-documented code ensure that others can replicate the steps and build on the outcomes of this project.

4 Result

This section presents the evaluation results for both BERT and LLAMA 3.2 1B models, assessed using accuracy, F1 score, precision, and recall. The performance metrics are summarized in Table 1, followed by confusion matrices that offer a detailed view of classification performance.

4.1 Performance Metrics

The overall performance of BERT and LLAMA 3.2 1B on the log anomaly detection task is shown in Table 1. Both models achieved excellent results, with LLAMA demonstrating slightly better performance across all metrics.

Model	Accuracy	F1 Score	Precision	Recall
BERT	0.998	0.981	0.986	0.976
LLAMA 3.2 1B	0.999	0.995	0.993	0.997

Table 1: Performance metrics for BERT and LLAMA 3.2 1B models.

Confusion matrices provide further insight into the classification performance of each model, including the counts of true positives, false positives, true negatives, and false negatives.

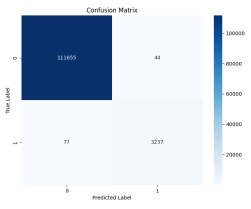

Figure 2: Confusion Matrix Bert

Figure 2 illustrates the confusion matrix for BERT, which achieved a strong accuracy of 99.8%. The model performed well across all metrics, particularly in precision (0.986).

Figure 3 shows the confusion matrix for LLAMA 3.2 1B, which achieved a slightly higher accuracy of 99.9%. LLAMA demonstrated superior recall (0.997), indicating its robustness in detecting anomalous log traces.

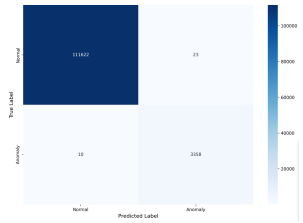

Figure 3: Confusion Matrix LLaMA

Overall, the results indicate that both models are highly effective for log anomaly detection, with LLAMA 3.2 1B offering marginally better performance due to its ability to generalize across a broader range of patterns.

5 Discussion

The results of this project highlight the comparative performance of BERT and LLAMA 3.2 1B in the context of log anomaly detection. LLAMA 3.2 1B demonstrated slightly better performance across all metrics, particularly in recall, which reflects its superior ability to identify anomalous log traces. This result aligns with expectations, as LLAMA's advanced architecture and larger context-handling capabilities make it well-suited for capturing the nuanced patterns present in log data. However, BERT also delivered strong results, showcasing its effectiveness as a reliable and efficient tool for log anomaly detection tasks.

The primary objectives of this project were to demonstrate the applicability of open-source LLMs for log anomaly detection, compare the strengths of BERT and LLAMA, and identify areas for improvement. These objectives were successfully addressed, with both models showing strong potential for automating log analysis. LLAMA's marginally better performance underscores the value of leveraging newer architectures, but the evaluation also highlights limitations stemming from the dataset itself, particularly its imbalance.

The unbalanced nature of the dataset likely influenced the models' performance, particularly in terms of generalization. This imbalance poses challenges by skewing model predictions toward the majority class, potentially impacting recall and precision for the minority class. Addressing this limitation presents an opportunity for future work, where strategies such as oversampling, undersampling, or synthetic data generation could be explored to improve generalization without introducing bias. A further avenue for exploration involves assess-

ing the robustness of these models across diverse datasets. Real-world log data can vary significantly in structure and content across different domains, making it crucial to evaluate whether the models trained in this project can generalize effectively to new datasets or require additional fine-tuning. This would provide valuable insights into their adaptability and resilience, particularly for applications in dynamic and heterogeneous environments.

Looking forward, transforming log data into alternative formats, such as time-series representations, offers the potential to enable real-time analysis. Models like the Temporal Fusion Transformer or Liquid Foundation Model, which are specifically designed for temporal dependencies, could be explored to further enhance anomaly detection accuracy and efficiency. These approaches are particularly relevant for practical applications in cybersecurity, where handling continuous data streams in real-time is essential. Adapting LLM architectures, such as LLAMA and BERT, to effectively process temporal data could provide another exciting direction for future work.

The findings from this project underscore the potential of open-source LLMs like LLAMA 3.2 1B and BERT as effective tools for log anomaly detection. Their adaptability, transparency, and reproducibility make them practical solutions for enhancing cybersecurity infrastructure. By addressing the limitations identified here and expanding the scope of future work, this project lays the groundwork for more resilient and adaptive anomaly detection systems that align with the evolving challenges of the field.

6 Related Works

Recent advancements in log anomaly detection have explored the application of state-of-the-art techniques, particularly leveraging Large Language Models (LLMs) for enhanced accuracy and adaptability. One such approach is LogGPT, which builds upon GPT-2 to predict the next token in a sequence, corresponding to the next expected "normal" log entry (Shang et al., 2023). LogGPT employs a statistical Top-K sampling method to make these predictions and integrates Proximal Policy Optimization (PPO), a reinforcement learning technique that adjusts policies based on gradient methods to maximize expected rewards (Schulman et al., 2017). By minimizing deviations between old and new policies, this method improves generalization and performance across various datasets.

LogGPT has demonstrated superior performance compared to earlier methods, particularly in its ability to identify anomalies with higher precision and recall. The integration of reinforcement learning and language modeling allows LogGPT to adapt dynamically to different log formats, making it a robust solution for log anomaly detection. These capabilities highlight the potential of transformer-based architectures and reinforcement learning to address the complexities of analyzing large-scale log data.

While LogGPT achieves promising results, it relies on GPT-2 as its foundation, and the specific reinforcement learning techniques used may introduce additional computational complexity. This makes it distinct from the approach presented in this project, which focuses on fine-tuning open-source LLMs such as BERT and LLAMA. These models, while not employing reinforcement learning, achieve comparable performance with simpler implementations and provide a baseline for further exploration.

By situating this project alongside approaches like LogGPT, it is evident that leveraging LLMs for log anomaly detection is a rapidly growing area. The findings of this project complement such advancements by emphasizing the applicability of fine-tuned models on labeled log data, while also addressing practical concerns like dataset imbalance and reproducibility.

7 Conclusion

This project demonstrated the effectiveness of Large Language Models, particularly BERT and LLAMA 3.2 1B, in the domain of log anomaly detection. LLAMA 3.2 1B outperformed BERT across key evaluation metrics, including accuracy, precision, recall, and F1 score, showcasing its ability to capture complex patterns within log sequences. These results underscore the potential of LLMs as valuable tools for automating anomaly detection tasks in cybersecurity.

The work also establishes a foundation for future advancements in this area. It will guide the development of a master's thesis aimed at addressing current limitations, such as data imbalance, through the creation of a diverse and comprehensive dataset. This dataset will facilitate more robust training and evaluation, enabling models to generalize effectively across dynamic environments.

Further exploration of advanced architectures and alternative approaches will also be prioritized to push the boundaries of log anomaly detection capabilities.

By building on these findings, future work can enhance the adaptability, reliability, and practicality of LLMs in real-world cybersecurity applications. Open-source models like BERT and LLAMA 3.2 1B offer not only strong performance but also transparency and reproducibility, making them suitable for addressing the challenges of modern cybersecurity infrastructure. The continued refinement of these models and supporting methodologies has the potential to significantly strengthen defenses against evolving threats, ensuring more secure and resilient systems.

Acknowledgments

Computer resources and research IT support were provided by UTS of the University of Iceland through the Icelandic Research e-Infrastructure project (IREI), funded by the Icelandic Infrastructure Fund.

References

Rishi Bommasani, Drew A Hudson, Ehsan Adeli, Russ Altman, Simran Arora, Sydney von Arx, Michael S Bernstein, Jeannette Bohg, Antoine Bosselut, Emma Brunskill, et al. 2021. On the opportunities and risks of foundation models. *arXiv preprint arXiv:2108.07258*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. 2017. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*.

Mengqi Shang, Shuhan Yin, Hao Gao, Hengfeng Zhao, Hao Jiang, Zheng Zhang, Mengqiang Li, and Yu Zhou. 2023. Loggpt: Log anomaly detection via gpt. *arXiv preprint arXiv:2309.14482*.

Konstantin Shvachko, Hairong Kuang, Sanjay Radia, and Robert Chansler. 2010. The hadoop distributed file system. In *2010 IEEE 26th symposium on mass storage systems and technologies (MSST)*, pages 1–10. IEEE.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. 2023. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 5998–6008.

Andy B. Yoo, Morris A. Jette, and Mark Grondona. 2003. Slurm: Simple linux utility for resource management. In *Proceedings of the 9th International Workshop on Job Scheduling Strategies for Parallel Processing (JSSPP)*, pages 44–60. Springer.