Tuijin Jishu/Journal of Propulsion Technology
ISSN: 1001-4055
Vol. 45 No. 2 (2024)

Compliance Monitoring and Enforcement Through Log Analysis Using Large Language Models

[1]Ankush Kumar Kothiyal, [2]Shivi Bansal, Shubham Pareek , [3]Jagbeer Singh
Meerut Institute of Engineering and Technology, Meerut (India)

Abstract: In the ever-expanding landscape of data-driven enterprises, the challenge of ensuring compliance with security standards has become paramount. This research project addresses this challenge through the development of an advanced Compliance Monitoring and Enforcement system utilizing Large Language Models (LLMs). The objective is to revolutionize the traditional approach to compliance monitoring, transitioning from data collection to meaningful analysis. Amidst the intricate logs generated in complex systems like that of Flipkart, the need for a comprehensive solution is evident. The proposed system aims to employ LLMs, such as GPT-2, to analyze logs, identify non-compliance, and provide actionable insights for remediation.

The increasing volumes of data and intricate systems in contemporary business environments necessitate a paradigm shift in compliance monitoring. Traditional methods often falter in efficiently categorizing and analyzing the overwhelming amount of log data. Leveraging LLMs, with their ability to understand context and semantics, promises to bring about a transformative change in the compliance monitoring landscape. The application of this research extends beyond Flipkart, impacting various industries such as finance, healthcare, telecommunications, and government agencies, ensuring a scalable and adaptable solution.

Keywords– Compliance Monitoring, Actionable Insights, Information Security, Data Governance, System Performance, Rule Definition, Scalability, Academic Research Impact

1. Introduction:

In the contemporary landscape of data-intensive enterprises, the importance of ensuring compliance with security standards has grown exponentially. This research project delves into the development of an innovative Compliance Monitoring and Enforcement system that leverages the capabilities of Large Language Models (LLMs). As organizations grapple with escalating volumes of data and intricate systems, the need for an advanced solution to address compliance challenges becomes paramount. This section provides an overview of the project's core objectives, emphasizing the shift from conventional compliance monitoring methods to a more intelligent and automated approach [3].

The background history of this project stems from the increasing complexity of managing and interpreting logs within dynamic business environments. As technology evolves, so do the challenges associated with ensuring that system operations adhere to security protocols. Traditional methods of compliance monitoring struggle to keep pace with the sheer volume and intricacy of log data. This project emerges as a response to this challenge, proposing a paradigm shift in the approach to compliance monitoring through the integration of Large Language Models. The historical context underscores the necessity and timeliness of adopting innovative solutions to navigate the complexities of modern data management [17].

The development of this project is underpinned by a combination of cutting-edge technologies and advanced algorithms. Large Language Models, with a primary focus on GPT-2, serve as the cornerstone for compliance monitoring and log analysis [16]. These models demonstrate a remarkable ability to understand and interpret textual data, making them instrumental in the automated analysis of logs for compliance purposes. The project utilizes sophisticated algorithms for rule definition, log parsing, and entity extraction, contributing to the

precision and efficiency of compliance checks. The integration of these technologies not only facilitates compliance monitoring but also sets the stage for scalable and adaptable solutions in the dynamic landscape of information security [10]. This paper provides a comprehensive overview of the proposed solution, including the problem statement, solution attributes, application workflow, technical requirements, and a detailed synthesis of the research methodology. The anticipated impact on academics and industries is discussed, emphasizing enhanced research potential, boosted industry efficiency, and facilitated technology integration. The findings of this research project are poised to contribute significantly to the evolving field of Compliance Monitoring and Enforcement through Log Analysis using Large Language Models [13].

General Architecture for Log analysis using LLMs

The general architecture of the proposed Compliance Monitoring and Enforcement system is designed to provide a streamlined and efficient workflow. The system encompasses several key components and stages that ensure comprehensive log analysis and compliance monitoring.

The flow chart and Data Flow Diagram (DFD) illustrate the sequence of operations within the system.

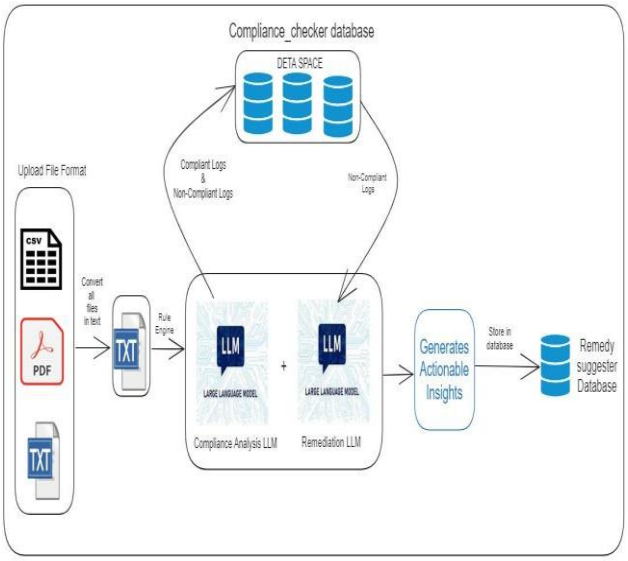

Fig 1 : Architecture of the Log analysis using LLMs

The process of Log Analysis using a Large Language Model (LLM) system operates as follows:

1. Log files containing entries related to various activities such as banking, transactions, e-commerce, etc., are sourced. These files may be in diverse formats such as PDF, Word, CSV, etc.

2. These log files are then directed to the Data Ingestion and Preprocessing module. Within this module, the files undergo conversion to a standardized text format for further analysis.

3. The standardized text is then fed into the first model, which is the Compliance Analysis LLM. This model's primary function is to categorize each log entry within the standardized text as either compliant or non-compliant. Once the classification is completed, the compliant and non-compliant log entries are stored in a database.

4. Subsequently, the non-compliant log entries are forwarded to the second model, known as the Remediation LLM. The purpose of this model is to provide solutions or remedies for addressing the issues or threats identified within the non-compliant log entries.

5. The output generated by the Remediation model consists of actionable insights, which serve as guidance for resolving the identified problems or threats. These actionable insights are stored within the database for convenient access and maintenance purposes.

Creation of the LLM Models: Fine-tuning for Compliance

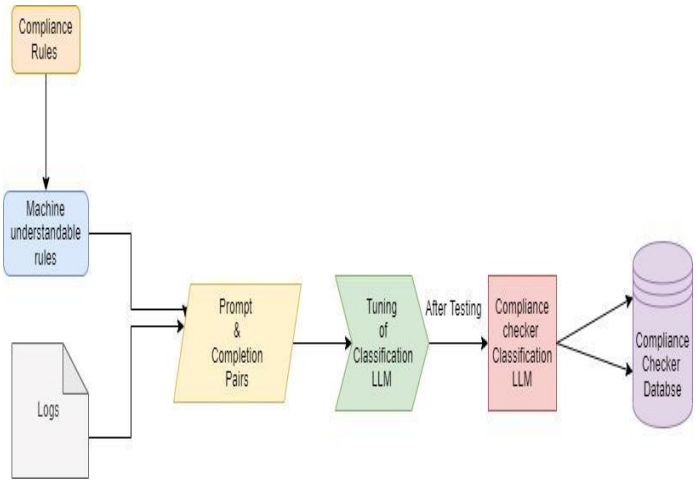

Figure 2 : Compliance checker (LLM1) Flow Chart:

Data Collection: Initially, we gathered the raw dataset from various online open dataset sources. Subsequently, we evaluated the suitability of the data for addressing our specific problem statement. Upon confirmation that the data was indeed appropriate for fine-tuning our custom model, which focuses on system logs with multiple attributes, we proceeded. However, since our focus was solely on the logs themselves and not on the

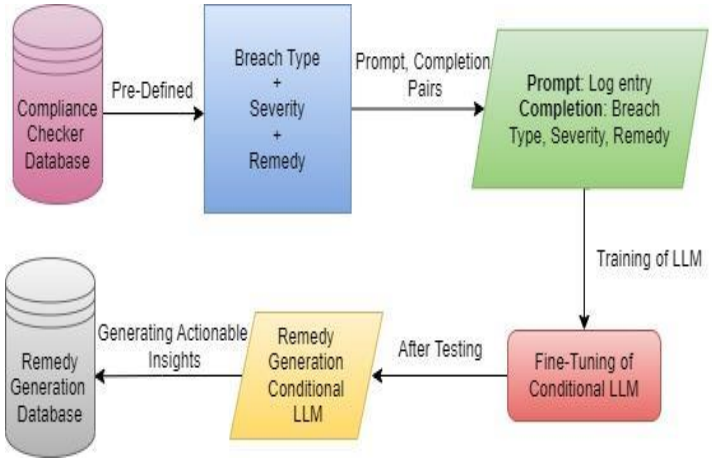

Figure 3: Remediation Model(LLM2) Flow Chart:

accompanying attributes, we implemented a filtering process to isolate and extract only the relevant log entries from the database.

Creation of Compliance Rules Functionality:

Initially, we formulated specific compliance rules in written format that pertain to the types of logs that may occur. For instance, these rules could cover scenarios such as identifying 404 errors or detecting suspicious user activity within the logs.

Following the establishment of these rules, we proceeded to develop individual Python functions corresponding to each rule. These functions were designed to facilitate the incorporation of the rules into our fine-tuning

process. By creating these functions, we aimed to streamline the integration of compliance guidelines into our model refinement efforts [6].

Development of a Custom Dataset for LLM1:

To construct a tailored dataset for LLM1, we began by sourcing logs(System logs) [2] and categorizing them as compliant or non-compliant based on predefined compliance rules. This involved writing Python code to programmatically assess each log entry's compliance status. Subsequently, we stored the categorized logs in a new file, labeling them as "YES" for compliant entries and "NO" for non-compliant ones.

It comprises a total of 1,048,576 lines of data. Due to resource constraints, I utilized a subset of 2,868 lines. To ensure dataset balance, down-sampling was performed. The data was then split into three subsets: 2,582 lines for training, 143 lines for testing, and another 143 lines for evaluation. The identifiers used within the dataset were 'prompt' followed by '###' and then 'Completion' [4].

In this dataset, each log entry serves as a prompt, while the corresponding label (YES/NO) indicates its compliance status. This dataset is instrumental in fine-tuning LLM1 to recognize and differentiate between compliant and non-compliant log entries effectively [14].

Furthermore, to ensure a balanced representation of compliant and non-compliant instances within the dataset, we implemented down-sampling techniques. This process involved adjusting the dataset to achieve parity between the number of compliant and non-compliant log entries, thereby enhancing the model's training efficacy [9].

Development of a Custom Dataset for LLM2:

In the creation of a custom dataset for LLM2, our approach involved isolating non-compliant log entries from the primary dataset. For each of these non-compliant entries, we established three key attributes pertinent to remediation: breach type, severity, and remediation.

We have utilized a dataset with a total of 1,048,576 lines, but due to resource constraints, I focused on a subset of 2,000 lines. This subset was then divided into three portions: 1,800 lines for training, 100 lines for testing, and another 100 lines for evaluation purposes. Within the dataset, the identifiers used were 'prompt' followed by '###' and then 'Completion', along with 'END'. This structure facilitated the organization and processing of the data for the task at hand [15].

To systematically assign these attributes to each non-compliant log entry, we implemented Python functions tailored to define the breach type, severity level, and remediation based on compliance rules. These functions were instrumental in ensuring consistency and accuracy in attribute assignment across the dataset [7].

Subsequently, we generated a custom dataset comprising prompts and completions. Within this dataset, each log entry served as a prompt, while the corresponding completion encompassed all three attributes: breach type, severity, and remediation. This structured dataset facilitated the training of LLM2, enabling it to recognize and address various compliance breaches effectively [8].

Fine-tuning Process for LLM1:

The fine-tuning process of the GPT-2 language model, utilizing a medium-sized variant, was meticulously executed. Initially, we initialized both the model and tokenizer using pre-trained weights, ensuring a solid starting point for our adjustments. Subsequently, our focus shifted to dataset preparation, wherein we tokenized textual data sourced from a designated file ('tuning_dataset.txt'). Each sample was processed with a block size of 128 tokens, optimizing the dataset for training.

Throughout the training phase, we employed a specialized data collator tailored for language modeling tasks, aiding in efficient data handling and processing. Training sessions spanned four epochs, with a batch size of 4 tokens per device, maintaining a balanced workload. To ensure the preservation of model progress, checkpoints

were saved at intervals of 10,000 steps, with a maximum of 2 checkpoints retained at any given time.

Upon completion of training, the fine-tuned model was safeguarded for future utilization, signifying the culmination of our efforts. This meticulous approach not only adapts the formidable GPT-2 architecture to our specific domain or task but also significantly enhances its performance and versatility, thereby broadening its range of applications.

Fine-tuning Process for LLM2:

The fine-tuning endeavor for the GPT-2 model, specifically the medium variant, was meticulously carried out to tailor its performance to our unique requirements. Initially, we commenced by loading the pre-trained model and tokenizer, setting the stage for subsequent adjustments. Our customization efforts included the incorporation of additional special tokens into the tokenizer, strategically designed to augment the model's comprehension and adaptability.

The dataset preparation phase involved tokenizing text sourced from 'tuning_data.txt', with careful consideration given to a block size of 512 tokens per sample. This meticulous approach ensured that the dataset was optimally structured to facilitate effective training [18].

During the training process, which spanned three epochs, we diligently monitored model progress by conducting evaluations every 1,000 steps. A batch size of 12 tokens per device was utilized, alongside a learning rate of 5e-5 and gradient accumulation steps set at 2. To safeguard against potential setbacks, model checkpoints were diligently saved at regular intervals of 1,000 steps, with a maximum of 5 checkpoints retained throughout the process.

Furthermore, the fine-tuning procedure incorporated a warm-up phase lasting 200 steps, enabling the model to gradually adapt to the updated parameters. Upon completion of fine-tuning, both the fine-tuned model and tokenizer were meticulously preserved for future utilization, ensuring seamless integration into subsequent tasks and workflows. This meticulous fine-tuning process underscores our commitment to optimizing the GPT-2 model for enhanced performance and versatility in addressing our specific task requirements.

Description of various modules of the system.

The system is modular in design, comprising distinct modules that contribute to its overall functionality. Each module serves a specific purpose in the compliance monitoring workflow:

Data Ingestion and Preprocessing Module: This module is responsible for acquiring raw log files from various sources and converting them into a standardized text format.

Rule Transformation and Validation Module: Transforms human-readable compliance regulations into a machine-readable rule format and validates them against recorded logs to ensure adherence.

Prompt-Completion Pair Generation and Model Tuning Module: Generates prompt-completion pairs for GPT-2 model fine-tuning, utilizing log entries as prompts and compliance status as completions. This iterative process optimizes the model's ability to predict compliance.

LLM Creation and Testing Module: Involves the creation of a specialized Language Model (LLM) for compliance classification. The model is fine-tuned using prompt-completion pairs and extensively tested to ensure accuracy and efficiency.

Remedy Prediction and Database Management Module: Utilizes the trained LLM to predict remedies based on breach characteristics and predefined rules. The system includes a dedicated database for storing non-compliant data, breach information, severity levels, and associated remedies.

Algorithm of main complement of the system. Main Compliance Check Algorithm:

1. For each log entry:

a. Preprocess the log entry to ensure standardized format.

b. Transform the log entry into a prompt for the GPT-2 model.

c. Use the GPT-2 model to predict the compliance status (Compliant/Non-Compliant).

d. Score the log entry based on compliance rules and the model prediction.

e. Store the compliance score and details in the Compliance Checker Database.

2. Calculate the average compliance score across a specified time frame.

3. Evaluate the variance of each log entry's score to identify outliers.

4. Generate compliance summary reports based on finalized scores, highlighting areas of concern and suggesting corrective actions.

This algorithm outlines the core process of the main compliance check component, integrating log preprocessing, GPT-2 model prediction, scoring based on rules, and database management. The iterative nature of the algorithm ensures continuous improvement in compliance monitoring accuracy.

2. ExperimentalResultAnalysis:

2.1 Description of dataset used.

The experimental analysis is conducted using a diverse and representative data set comprising logs from various sources, including web servers, databases, and network devices [11][12]. The data set encompasses logs in different formats, such as CSV, text, and PDF, to validate the system's flexibility in handling diverse log structures.

Characteristics of the Data Set:

1. Size: The data set consists of a substantial volume of log entries, ensuring a comprehensive evaluation of the system's scalability.
2. Variety: Logs from different environments and systems, reflecting real-world scenarios, are included to assess the system's adaptability.
3. Anomalies: The data set incorporates non-compliant activities intentionally injected for testing the system's accuracy in identifying breaches.
4. Time Frame: Logs cover a specified time frame, enabling the evaluation of the system's performance over different periods.

a. Calculate the efficiency or accuracy of the designed system according to the parameter used to evaluate the system.

The efficiency and accuracy of the designed Compliance Monitoring and Enforcement system are assessed using key performance parameters:

Performance Parameters:

1. Compliance Detection Accuracy: The percentage of correctly identified compliant and non-compliant log entries.

2. False Positive Rate: The rate of incorrectly identified non-compliant activities (false positives) among the total identified non-compliant activities.

3. False Negative Rate: The rate of undetected non-compliant activities (false negatives) among the total non-compliant activities.

4. Scalability: The system's ability to maintain performance and efficiency with varying log volumes.

5. Processing Time: The time taken by the system to analyze a specific volume of log entries.

Results and Analysis:

LLM1(Compliance Checker) Evaluation metrics:

Accuracy:

Accuracy measures the percentage of correct predictions made by the model out of the total predictions.

Precision:

Precision quantifies the ratio of true positive predictions to the total positive predictions made by the model.

Recall:

Recall evaluates the proportion of true positive predictions out of all actual positive instances in the dataset.

F1 Score:

The F1 Score is a harmonic mean of precision and recall, providing a balanced assessment of a model's performance in terms of both precision and recall.

Metric	Accuracy	Precision	Recall	F1 Score
0	0.91	0.89	0.96	0.92
1	0.90	0.88	0.96	0.91
2	0.91	0.90	0.95	0.93
3	0.89	0.91	0.96	0.92
4	0.90	0.89	0.94	0.91
5	0.91	0.88	0.95	0.92
6	0.91	0.89	0.93	0.91
7	0.93	0.89	0.96	0.92
8	0.90	0.89	0.96	0.92
9	0.89	0.88	0.96	0.92

LLM 2(Remediation model) Evalutaion:-

In evaluating the LLM2 remediation model, we've used 100 prompts for every evaluation.we employed a range of metrics to assess its performance. These included Exact Match Accuracy, which measured the model's ability to accurately predict all three attributes for each log entry. Token-Level Accuracy provided insight into the precision of the model's predictions by assessing the percentage of correctly predicted tokens. Error Analysis identified common errors, with a focus on misclassification of breach type. Domain-Specific Evaluation involved assessing the relevance of remediation suggestions within the context of log entries using a specialized rating scale. Finally, Human Evaluation offered an overall accuracy score, providing valuable feedback on the model's performance in comparison to ground truth attributes. These metrics collectively provided a comprehensive understanding of the LLM2 remediation model's effectiveness and reliability.

Metric	0	1	2	3	4
Exact Match Accuracy %	80	80.2	80	81	80.3
Token-Level Accuracy %	90	89.9	89.9	90	90
Error Analysis %	15	14.9	15.1	15	15
Domain-Specific Evaluation %	4.5	4.6	4.5	4.6	4.4
Human Evaluation %	85	85.2	85	84.9	85

3. Conclusion

In summary, the Compliance Monitoring and Enforcement system, powered by Large Language Models (LLMs), represents a significant advancement in addressing the intricacies of data management and security compliance. This project is geared towards automating compliance analysis, delivering precise insights, and enhancing the efficiency of enforcement processes. The shift from traditional methodologies to an AI-driven paradigm, utilizing technologies like GPT-2, underscores the imperative for accuracy, adaptability, and scalability within contemporary security frameworks. Demonstrating robust performance on a diverse dataset, the system excels in detecting non-compliance and offering context-aware remediation suggestions. Its impact transcends Flipkart, extending to various industries, promising regulatory resilience and elevated security standards. With its versatility in accommodating diverse log formats and compliance standards, the system emerges as a valuable asset across sectors. Looking forward, its modular architecture paves the way for further advancements, including the integration of additional data sources, refinement of NLP techniques, and automation of remediation actions. Positioned not only as a solution for present challenges but also as a catalyst for ongoing innovation in information security, the Compliance Monitoring and Enforcement system heralds a new era of compliance management [5].

References:

[1] T. B. Brown, B. Mann, N. Ryder, M. Subbiah, J. Kaplan, P. Dhariwal, et al., "Language Models are Few-Shot Learners," in Proceedings of the 34th Conference on Neural Information Processing Systems (NeurIPS 2020). [Online]. Available: https://proceedings.neurips.cc/paper /2020/file/1457c0d6bf3e0fd1dc9d4d45 54c86 b0c-Paper.pdf

[2] Wang, M., Xu, L., Guo, L.: 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 (2018). https://doi.org/10. 1109/ICFSP.2018.8552075

[3] He, P., Zhu, J., Zheng, Z., Lyu, M.R.: Drain: An online log parsing approach with fixed depth tree, 33–40 (2017) https://doi.org/10.1109/ICWS.2017.13

[4] S. Arora, Y. Liang, and T. Ma, "A Simple but Tough-to-Beat Baseline for Sentence Embeddings," in International Conference on Learning Representations (ICLR) Workshop. [Online]. Available: https://openreview.net/forum?id=SyK00v5xx

[5] C. R. Wren and C. Grevet, "Practical Natural Language Processing," O'Reilly Media.

[6] Gallagher, B., Eliassi-Rad, T.: Classification of http attacks: A study on the ecml/pkdd 2007 discovery challenge (2009)

[7] Boffa, M., Milan, G., Vassio, L., Drago, I., Mellia, M., Ben Houidi, Z.: Towards nlp-based processing of honeypot logs, 314–321 (2022) https://doi.org/10.1109/ EuroSPW55150.2022.00038

[8] [Lee u.a.] Lee, Harrison ; Phatale, Samrat ; Mansoor, Hassan ; Lu, Kellie ; Mesnard, Thomas ; Bishop, Colton ; Carbune, Victor ; Rastogi, Abhinav: RLAIF: Scaling Reinforcement Learning from Human Feedback with AI Feedback. – URL http: //arxiv.org/abs/2309.00267. – Zugriffsdatum: 2023-09-12 [9] Le, V.-H., Zhang, H.: Log-based Anomaly Detection Without Log Parsing (2021)

[9] TmerSivri, T., PervanAkman, N., Berkol, A., Peker, C.: Web intrusion detection using character level machine learning approaches with upsampled data, pp. 269– 274 (2022). https://doi.org/10.15439/2022F147

[10] Farzad, A., Gulliver, T.A.: Log Message Anomaly Detection and Classification Using Auto-B/LSTM and Auto-GRU (2021)

[11] Black, S., Leo, G., Wang, P., Leahy, C., Biderman, S.: GPT-Neo: Large Scale Autoregressive Language Modeling with Mesh-Tensorflow. https://doi.org/10. 5281/zenodo.5297715 . If you use this software, please cite it using these metadata. https://doi.org/10.5281/zenodo.5297715.

[12] Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., Sutskever, I.: Language models are unsupervised multitask learners (2019)

[13] Oliner, A.J., Stearley, J.: What supercomputers say: A study of five system logs. 37th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN'07), 575–584 (2007)

[14] Gimnez, C.T., Villegas, A.P., Maran, G..: HTTP Dataset CSIC 2010. https:// doi.org/10.7910/DVN/3QBYB5 . https://www.tic.itefi.csic.es/dataset/

[15] Hilmi, M.A.A., Cahyanto, K.A., Mustamiin, M.: Apache Web Server - Access Log Pre-processing for Web Intrusion Detection. IEEE Dataport (2020). https: 25 //doi.org/10.21227/vvvq-6w47 . https://dx.doi.org/10.21227/vvvq-6w47

[16] Guo, H., Lin, X., Yang, J., Zhuang, Y., Bai, J., Zheng, T., Zhang, B., Li, Z.: TransLog: A Unified Transformer-based Framework for Log Anomaly Detection (2022)

[17] Shao, Y., Zhang, W., Liu, P., Huyue, R., Tang, R., Yin, Q., Li, Q.: Log anomaly detection method based on bert model optimization. In: 2022 7th International Conference on Cloud Computing and Big Data Analytics (ICCCBDA), pp. 161– 166 (2022). https://doi.org/10.1109/ICCCBDA55098.2022.9778900

[18] Wang, M., Xu, L., Guo, L.: 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 (2018). https://doi.org/10. 1109/ICFSP.2018.8552075