Log Based Anomaly Detection using RoBERTa for Feature Extraction

Supriya Dara
School of Electrical Engineering and
Computer Science (EECS)
University of Ottawa
Ottawa, Canada
sdara100@uottawa.ca

***Abstract*—Log files in software-driven systems are critical for monitoring and diagnosing issues but are often voluminous and complex, making manual analysis impractical. Traditional log anomaly detection methods, relying on rule-based or statistical approaches, struggle with the variability and complexity of modern system logs. This paper proposes a novel methodology that leverages the advanced capabilities of the RoBERTa language model for robust feature extraction. Utilizing RoBERTa, our approach transforms unstructured log data into semantic vectors, enhancing the detection capabilities of machine learning algorithms. We detail the process of log parsing, feature extraction, and anomaly detection using a dataset from the Hadoop Distributed File System (HDFS) to evaluate our model. Our methodology outperforms traditional approaches like term frequency-inverse document frequency (TF-IDF) across various metrics, achieving near-perfect classification scores in detecting anomalous logs. The results demonstrate the effectiveness of integrating pre-trained language models in log anomaly detection, suggesting a scalable and efficient solution for handling complex log data in distributed systems. This study advances log anomaly detection and sets a precedent for applying natural language processing techniques in system monitoring.**

Keywords*—Log Anomaly Detection, Log Parsing, Feature Extraction, Pre-trained Language Models, RoBERTa, Machine Learning

I. Introduction

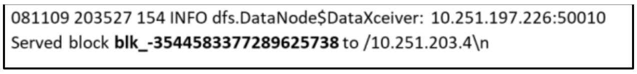
Software-intensive systems typically produce many logs that record different executions of these systems. A log file usually contains a rich source of information from normal and abnormal system operations (e.g., starting or stopping processes, allocating system resources, system crashes, granting user permissions, etc.) that can be used for troubleshooting, service maintenance, or security purposes [1]. Due to the potentially huge volume of logs, manual log analysis for anomaly detection can be time-consuming and error-prone [2].

Traditional methods for anomaly detection in log data often rely on manual rule-based approaches or statistical techniques, which may need help to capture the complexity and variability of modern system behaviors [3][4]. These methods may fail to detect attacks or anomalies involving logs from different systems or logs exhibiting different formats and structures [2].

Recent advancements in natural language processing (NLP) and machine learning (ML) have introduced innovative approaches to log anomaly detection [3][5]. While pre-trained language models have emerged as promising tools for enhancing the effectiveness of anomaly detection systems [6][7]. This paper addresses the need for automated and scalable anomaly detection solutions by proposing a novel approach to log anomaly detection. It leverages state-of-the-art pre-trained language models, such as RoBERTa [13], for feature extraction.

RoBERTa, a transformer-based language model pre-trained on vast corpora of text data, can encode rich contextual information, enabling more nuanced representations of log messages [8]. Utilizing pre-trained language models like RoBERTa for feature extraction can lead to effective ways to improve log anomaly detection [5][6][14].

In this study, we utilized a dataset from the Hadoop Distributed File System (HDFS) [10]. Fig.1 shows an example of a log entry from the dataset. This entry logs a data transfer event, providing the timestamp (081109 203527), indicating when the event occurred, the log level (INFO), denoting the severity or nature of the event, and a descriptive message. The message details that the data node 10.251.197.226 on port 50010 served a data block identified by blk_-3544583377289625738 to another node at address 10.251.203.4. Such entries, with their precise record of file operations and interactions within the HDFS, are integral to our analysis.



```
081109 203527 154 INFO dfs.DataNode$DataXceiver: 10.251.197.226:50010
Served block blk_-3544583377289625738 to /10.251.203.4\n
```

Fig. 1. Example of a Log Entry from the HDFS Dataset

Our approach involves several key steps:

1) **Log Parsing:** Raw unstructured logs are parsed into structured event templates using Drain [9], enabling the extraction of meaningful log features.
2) **Feature Extraction:** The parsed log event templates are converted into log sequences, which are then transformed into embedding vectors using RoBERTa. Semantic vectors are obtained through concatenation by considering the position and segment information of tokens in the log sequence.
3) **Anomaly Detection:** Semantic representations of log sequences are used as input features for various traditional ML algorithms to detect anomalies. Supervised learning methods are employed to learn from the semantic representations of normal and anomalous logs.

The order of words in a log sequence often influences the presence of anomalies. Consequently, anomalous log sequences differ significantly from normal ones, enabling effective classification through supervised learning approaches. Experimental results have demonstrated the superiority of our proposed method over traditional approaches, such as term frequency-inverse document frequency (TF-IDF), particularly in terms of accuracy and generalization.

We evaluated our method HDFS dataset [10], achieving the highest performance for anomalous logs classification, with an F1-score of 0.99. These results highlight the potential of RoBERTa embeddings to provide more robust and accurate solutions for log anomaly detection in complex distributed systems.

In the subsequent sections of this paper, we delve into the methodology, experimental setup, results, and discussions of our log anomaly detection framework. Through this study, we aim to analyze the strengths and limitations of utilizing pre-trained models for feature extraction, thereby contributing valuable insights and potential enhancements to the field of log anomaly detection.

II. Related Work

Numerous studies have investigated log anomaly detection using various heuristic and machine-learning methodologies. Notably, Song Chen & Hai Liao [5] addressed the challenge of anomaly detection in large-scale computer systems, a task complicated by the difficulty of classifying anomalies from system logs. The study proposes BERT-Log, a novel approach that treats log sequences as natural language sequences. While this method reduced the need for detailed log parsing, it struggled with huge datasets where model scalability became crucial. The study needs to fully address the computational demands and potential performance when scaling to real-world, large-scale environments. However, this approach has set a precedent in the domain, highlighting the potential of language models in log analysis.

J. K. S and A. B. [14] introduced a novel approach to phishing URL detection, utilizing RoBERTa for feature extraction and LSTM for classification. Unlike previous studies relying on heuristic and traditional machine learning methods, this research advances state-of-the-art by incorporating state-of-the-art models and addressing limitations such as dataset size and detection challenges. The study gives promising results in using language models for feature extraction.

Mvula et al. [12] introduced a novel approach that harnesses the power of robust transformer models to sift through heterogeneous log data. The study leverages transformers' adaptive capabilities to not only deal with the diverse nature of log files but also detect subtle anomalies that often go unnoticed with conventional methods. While their methodology significantly improves the detection of subtle anomalies, the diversity of log files poses a challenge to the consistency of performance across different system logs, which may need to be uniformly structured.

In another work by Lee et al. [7] introduced LAnoBERT, a parser-free system log anomaly detection method using BERT. This approach reduces dependency on pre-defined templates, which is beneficial for handling unstructured log data. However, the lack of structured parsing can sometimes lead to misinterpretations of log content, especially in logs with high variability in message formats.

B. Yu, et al. [15], discussed the efficacy and computational efficiency of deep learning methods in log anomaly detection. They noted that simpler algorithms often outperform more complex deep learning methods, which can be attributed to the sometimes unnecessary complexity and higher computational costs of deep learning models, particularly when dealing with simple or less noisy datasets.

He et al. [9] introduced Drain, an online log parsing tool that significantly reduces manual effort. While Drain automates the extraction of structured information, its fixed-depth tree approach might only capture some nuances in logs with highly variable or novel event types, potentially leading to information loss in diverse logging environments. This approach significantly reduced the manual effort required for log parsing and has been widely adopted.

Similarly, He et al. [16] and H. Li and Y. Li [21] both highlight the importance of tailored feature extraction methods in enhancing anomaly detection. However, these approaches often necessitate extensive customization to adapt to specific characteristics of different distributed systems, a resource-intensive process that may not always be feasible in less controlled environments.

III. Methodology

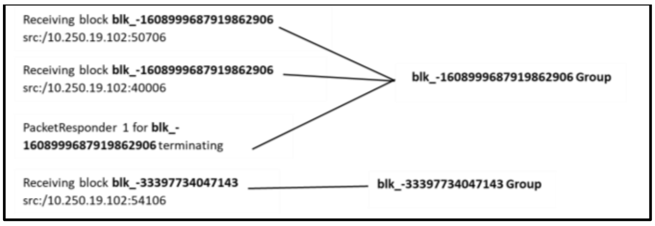
This study's methodology involves several key components: data preparation, log parsing, feature extraction, model selection, and evaluation. Each step is designed to process unstructured log data, extract meaningful features, and train various ML models for anomaly detection. Below, we outlined the methodology in detail.

A. Data Preprocessing

In this methodology we used the Hadoop Distributed File System (HDFS) log data collected by Xu et al. [11] in a private cloud environment using performance tests and manually labelled it using handcrafted rules to identify the anomalies in the data. This dataset comprises 11,175,629 log messages, divided into sequences using block IDs and labelled as normal or abnormal.

The HDFS log data is split into training and testing sets to facilitate model training and evaluation. The splitting is performed sequentially, with the first 80% of logs allocated to the training set and the remaining 20% to the testing set. Sequential splitting is chosen over random splitting to preserve the temporal nature of the log data, ensuring that event sequences remain intact for both training and testing purposes.

Each log message in the HDFS dataset is associated with a unique block ID, representing a distinct data block within the file system. These block IDs serve as identifiers for grouping related log events. Fig. 2 shows an example of HDFS log data in which the first three logs are part of one block ID, and the fourth is part of a different block ID.



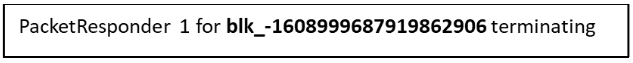
```
Receiving block blk_-1608999687919862906
src:/10.250.19.102:50706

Receiving block blk_-1608999687919862906
src:/10.250.19.102:40006

PacketResponder 1 for blk_-
1608999687919862906 terminating

Receiving block blk_-33397734047143
src:/10.250.19.102:54106
```

Fig. 2. Grouping of HDFS log data with identical Block IDs

During the splitting process, we ensured that all sequences of log events corresponding to individual block IDs are assigned to the training or testing set. This approach prevents fragmentation of event sequences across different sets, enabling more accurate model evaluation.

B. Log Parsing

Log messages are typically unstructured, consisting of log event templates and variables within each message. For example, a log message may appear as shown in Fig. 3.



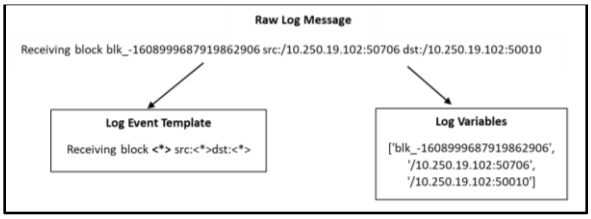
```
PacketResponder 1 for blk_-1608999687919862906 terminating
```

Fig. 3. Example of Raw Log Message

Traditional methods for parsing log files involve manual, hand-crafted statements based on domain knowledge. However, these approaches are time-intensive and require constant modification to accommodate changes in log formats [20]. Automated log parsing aims to streamline this process by reducing the manual effort.

In this study, the Drain automatic log parser, available through the Logparser toolkit [9], is utilized for automated log parsing. The Drain log parser automates the parsing of raw unstructured HDFS log data. This tool utilizes sophisticated algorithms to identify event templates and variable components within each log message.

The raw unstructured HDFS log data is parsed using Drain to generate structured data comprising log event templates and variables, as illustrated in Fig. 4. This parsing process identifies 46 unique event templates, denoted as E1 to E46 for the study.



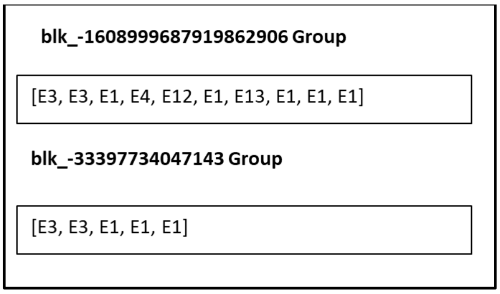
```
Receiving block blk_-1608999687919862906 src:/10.250.19.102:50706 dst:/10.250.19.102:50010
```

```
Receiving block <*> src:<*>dst:<*>
```

```
['blk_-1608999687919862906',
'/10.250.19.102:50706',
'/10.250.19.102:50010']
```

Fig. 4. Parsing of an Unstructured Log Message into Template and Variables.

Each event template captures a specific type of log event observed in the dataset. After extracting log variables, the parsed logs are grouped based on their HDFS block IDs as shown in Fig. 5. This grouping facilitates the creation of sequences of log events for each HDFS block ID, enabling the structured and organized analysis of log data.



blk_-1608999687919862906 Group

[E3, E3, E1, E4, E12, E1, E13, E1, E1, E1]

blk_-33397734047143 Group

[E3, E3, E1, E1, E1]

Fig. 5. Grouping Parsed Logs based on their block IDs

C. Feature Extraction

Feature extraction involves converting structured log data into numerical representations that capture underlying patterns and characteristics of log events. Traditional feature extraction techniques, such as TF-IDF, provide a basic representation of log messages [21]. However, leveraging advanced language models like RoBERTa allows for extracting richer, contextualized features.

Each log message is tokenized into a sequence of tokens, and RoBERTa embeddings are generated for each token. These embeddings capture the semantic meaning of each token in the context of the entire log message. To obtain a fixed-dimensional representation for each log message, aggregation techniques like max-pooling are applied over token embeddings to get a single vector representation for the entire log message.

RoBERTa-based feature vectors serve as input features for the log anomaly detection model, replacing traditional features like TF-IDF. By integrating RoBERTa into feature extraction, the log anomaly detection system can leverage the model's contextualized embeddings to capture complex relationships within log messages.

Subsequently, feature extraction is conducted for each HDFS block ID sequence of events through a three-step process. Firstly, event counts are computed for each block ID grouping, considering the significance of event templates in the entire dataset. Secondly, sliding window event counts are applied to capture the sequential history of events within each block ID, thereby generating matrices representing the temporal evolution of log events within a block ID. Lastly, the sliding window event count matrices are multiplied by the corresponding block ID embeddings, resulting in matrices based on RoBERTa contextualized embeddings instead of event counts. Fig. 6 outlines the feature extraction process for log data using the RoBERTa model, from raw events to contextualized feature vectors.

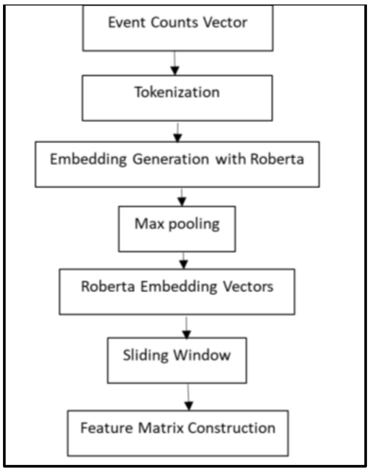

Event Counts Vector
Tokenization
Embedding Generation with Roberta
Max pooling
Roberta Embedding Vectors
Sliding Window
Feature Matrix Construction

Fig. 6. Feature extraction using RoBERTa

D. Model Training

The feature matrices generated using RoBERTa serve as an input to the training model. In the model training phase, various machine learning algorithms, including Support Vector Machine (SVM) [17][19], Random Forest (RF) [18], Decision Tree (DT) [22], and Gradient Boosting Machine (GBM) [23], are employed to classify normal and anomalous log events using RoBERTa-based input features.

E. Model Evaluation

The methodology includes a comparative analysis of our proposed RoBERTa-based method for feature extraction against traditional methods like TF-IDF on various ML models. The performance of the various ML models, such as SVM, RF, DT, LR, and GBM, is evaluated using precision, recall, and F-score metrics on both the training and testing datasets.

IV. Experiment

In this section, we outline the experimental setup utilized to gauge the efficacy of our model for log-based anomaly detection.

- **Dataset:** We employed a meticulously labelled dataset comprising 11,175,629 logs obtained from LogHub. Among these logs, 436,139 Block IDs were categorized as normal (class 0), while 11,808 Block IDs were classified as anomalies (class 1).
- **Log Parsing:** Prior to model implementation, the raw log data underwent a parsing procedure by implementing the Drain log parser, a tool available through the Logparser toolkit to structure it into meaningful log event templates and variables.
- **Feature Extraction:** Our model, developed using PyTorch, utilized the RoBERTa model to extract features and leverage its contextualized embeddings.
- **Training:** The dataset was split into training and testing sets, with an 80:20 ratio for training and testing, respectively. Various machine learning algorithms were trained on the training data to explore the model's versatility and adaptability.
- **Evaluation Metrics:** To calculate our model's performance, we evaluated it using standard metrics, including Accuracy, Precision, Recall, and F1 Score.

V. Result and Discussion

Our experimentation with various machine learning models utilizing RoBERTa for feature extraction yielded promising results in log-based anomaly detection. Table 1 presents the performance metrics of different models, including DT, GBM, SVM, and RF. Notably, all models achieved exceptional accuracy, precision, recall, and F1 scores, demonstrating the efficacy of RoBERTa-based feature extraction in capturing nuanced patterns within log data.

TABLE I. Performance Metrics of Different Models Using RoBERTa for Feature Extraction

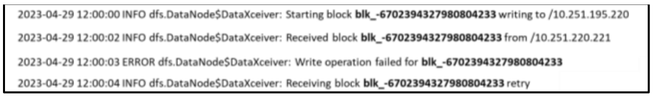
Models	**Performance Metrics**			
	Accuracy	***Precision***	***Recall***	***F1-score***
DT	0.9990	1.00	0.97	0.98
GBM	0.9991	1.00	0.97	0.98
SVM	**0.9998**	1.00	0.99	1.00
RF	**0.9998**	**1.00**	**1.00**	**1.00**

Furthermore, a comparative analysis between our proposed method and the traditional approach using TF-IDF for feature extraction is presented in Table 2. The results underscore the superiority of our proposed method, which outperforms the TF-IDF approach across all performance metrics. The proposed method achieved near-perfect accuracy, precision, recall, and F1-score, indicating its robustness in accurately identifying anomalous log events.

TABLE II. Comparative Analysis of Our Proposed Method and the TF-IDF Approach for Feature Extraction

Methods	**Performance Metrics**			
	Accuracy	***Precision***	***Recall***	***F1-score***
Method using TF-IDF	0.9985	0.99	0.99	0.99
Proposed Method	**0.9998**	**1.00**	**1.00**	**1.00**

To further demonstrate the advantages of RoBERTa over traditional methods, we analyzed a specific challenging log sequence illustrated in Fig. 7, in which TF-IDF faltered. Table 3. illustrates the results for this log sequence, which underscores the benefits of leveraging advanced language models to enhance the accuracy and reliability of log-based anomaly detection systems.



```
2023-04-29 12:00:00 INFO dfs.DataNode$DataXceiver: Starting block blk_-6702394327980804233 writing to /10.251.195.220
2023-04-29 12:00:02 INFO dfs.DataNode$DataXceiver: Received block blk_-6702394327980804233 from /10.251.220.221
2023-04-29 12:00:03 ERROR dfs.DataNode$DataXceiver: Write operation failed for blk_-6702394327980804233
2023-04-29 12:00:04 INFO dfs.DataNode$DataXceiver: Receiving block blk_-6702394327980804233 retry
```

Fig. 7. Example log sequence from the HDFS dataset

TABLE III. Performance Comparison for One Specific Log Sequence

Methods	***Detection***	***Precision***	***Recall***	***F1-score***
Method using TF-IDF	Normal	0.99	0.99	0.99
Proposed Method	**Anomaly**	**1.00**	**1.00**	**1.00**

a) *Traditional Method (TF-IDF) Analysis*: Identified the sequence as normal based on individual log entries term frequencies, missing the anomalous pattern due to lack of context.

b) *RoBERTa Method Analysis*: Correctly flagged the sequence as anomalous by understanding the contextual relationship between rapid succession and unusual ordering of operations, which is atypical for standard user behavior.

The results of our experiments highlight the effectiveness of leveraging RoBERTa for feature extraction in log anomaly detection. The superior performance of our proposed method, as evidenced by the significantly higher accuracy, precision, recall, and F1-score compared to the TF-IDF approach, underscores the importance of utilizing advanced language models in log analysis tasks.

The exceptional accuracy and precision achieved by all models, particularly SVM and RF, indicate their capability to accurately classify both normal and anomalous log events. The high recall scores further emphasize the model's ability to effectively identify anomalous log events while minimizing false negatives.

Moreover, the near-perfect F1-scores attained by our proposed method validate its robustness and effectiveness in capturing complex relationships within log messages. The utilization of RoBERTa embeddings enables the models to extract rich contextualized features, thereby enhancing their ability to discern anomalies amidst vast amounts of log data.

Our findings suggest that integrating advanced language models like RoBERTa into log anomaly detection systems can significantly improve their performance and reliability. This approach not only simplifies the feature extraction process but also enhances the models' ability to generalize and adapt to diverse log datasets, thereby paving the way for more effective and scalable log analysis solutions.

VI. Conclusion

In this study, we proposed a novel approach to log anomaly detection leveraging state-of-the-art pre-trained language models, specifically RoBERTa, for feature extraction. Through a comprehensive methodology encompassing data preparation, log parsing, feature extraction, model training, and evaluation, we demonstrated the effectiveness of RoBERTa-based feature extraction in enhancing log anomaly detection systems. Our experimental results showcase the superior performance of various machine learning models, including Decision Tree, Gradient Boosting Machine, Support Vector Machine, and Random Forest, when utilizing RoBERTa embeddings for feature extraction. The achieved accuracy, precision, recall, and F1-scores underscore the robustness and reliability of our proposed method in accurately identifying anomalous log events. Comparative analysis with traditional methods like TF-IDF further validates the efficacy of our approach, highlighting its superiority across all performance metrics. Moreover, our findings underscore the significance of integrating advanced language models like RoBERTa into log anomaly detection systems, enabling them to capture complex relationships within log messages and adapt to diverse log formats and structures. This approach simplifies the feature extraction process and enhances the models' ability to generalize and scale across various log datasets and system environments.

Through this study, we contribute valuable insights and potential enhancements to the field of log anomaly detection, emphasizing the transformative impact of pre-trained language models in addressing the challenges associated with log data analysis. Future research directions may involve exploring the application of other advanced language models and investigating ensemble learning techniques to improve the performance and adaptability of log anomaly detection systems.

Acknowledgment

I would like to thank my supervisor, Miguel Garzon, an Adjunct Professor at the University of Ottawa, for his invaluable guidance and unwavering support throughout this project.

References

[1] Max Landauer, Sebastian Onder, Florian Skopik, Markus Wurzenberger, "Deep learning for anomaly detection in log data: A survey," Machine Learning with Applications, Volume 12, 2023.

[2] V. -H. Le and H. Zhang, "Log-based Anomaly Detection Without Log Parsing," 2021 36th IEEE/ACM International Conference on Automated Software Engineering (ASE), Melbourne, Australia, 2021, pp. 492-504.

[3] Y. Wang and X. Li, "FastTransLog: A Log-based Anomaly Detection Method based on Fastformer," 2022 9th International Conference on Dependable Systems and Their Applications (DSA), Wulumuqi, China, 2022, pp. 446-453.

[4] P. Ryciak, K. Wasielewska, A. Janicki, "Anomaly Detection in Log Files Using Selected Natural Language Processing Methods," Applied Sciences, 2022.

[5] Song Chen & Hai Liao, "BERT-Log: Anomaly Detection for System Logs Based on Pre-trained Language Model," Applied Artificial Intelligence, 2022.

[6] Guo, Haixuan & Yuan, Shuhan & Wu, Xintao, "LogBERT: Log Anomaly Detection via BERT," 2021.

[7] Lee, Yukyung & Kim, Jina & Kang, Pilsung. "LAnoBERT: System Log Anomaly Detection based on BERT Masked Language Model," 2021.

[8] Ramazan Mengi, Hritik Ghorpade, Arjun Kakade, "Fine-tuning T5 and RoBERTa Models for Enhanced Text Summarization and Sentiment Analysis," The Great Lakes Botanist, 2023.

[9] Pinjia He, Jieming Zhu, Zibin Zheng, and Michael R. Lyu, "Drain: An Online Log Parsing Approach with Fixed Depth Tree," Proceedings of the 24th International Conference on Web Services (ICWS), 2017.

[10] Jieming Zhu, Shilin He, Pinjia He, Jinyang Liu, Michael R. Lyu, "Loghub: A Large Collection of System Log Datasets for AI-driven Log Analytics," IEEE International Symposium on Software Reliability Engineering (ISSRE), 2023.

[11] Wei Xu, Ling Huang, Armando Fox, David Patterson, Michael Jordan, "Detecting Large-Scale System Problems by Mining Console Logs," in Proc. of the 22nd ACM Symposium on Operating Systems Principles (SOSP), 2009.

[12] P.K. Mvula, P. Branco, GV. Jourdan, H.L. Viktor, "HEART: Heterogeneous Log Anomaly Detection Using Robust Transformers," In: Bifet, A., Lorena, A.C., Ribeiro, R.P., Gama, J., Abreu, P.H. (eds) Discovery Science. DS 2023.

[13] Y. Liu et al., "RoBERTa: a robustly optimized BERT pretraining approach," ArXiv abs/1907.11692 (2019).

[14] J. K. S and A. B, "Phishing URL detection by leveraging RoBERTa for feature extraction and LSTM for classification," 2023 Second International Conference on Augmented Intelligence and Sustainable Systems (ICAISS), Trichy, India, 2023, pp. 972-977.

[15] B. Yu et al., "Deep Learning or Classical Machine Learning? An Empirical Study on Log-Based Anomaly Detection," in 2024 IEEE/ACM 46th International Conference on Software Engineering (ICSE), Lisbon, Portugal, 2024 pp. 392-404.

[16] S. He, J. Zhu, P. He and M. R. Lyu, "Experience Report: System Log Analysis for Anomaly Detection," 2016 IEEE 27th International Symposium on Software Reliability Engineering (ISSRE), Ottawa, ON, Canada, 2016, pp. 207-218.

[17] L. Wang, "Support vector machines: theory and applications," Springer Science & Business Media, 2005.

[18] L. Breiman, "Random forests," Machine learning, 45, 5–32, Springer, 2001.

[19] Y. Liang, Y. Zhang, H. Xiong, and R. Sahoo, “Failure prediction in IBM BlueGene/L event logs,” in Proc. 7th IEEE Int. Conf. Data Mining (ICDM), 2007, pp. 583–588.

[20] Jieming Zhu, Shilin He, Jinyang Liu, Pinjia He, Qi Xie, Zibin Zheng, Michael R. Lyu, "Tools and Benchmarks for Automated Log Parsing," International Conference on Software Engineering (ICSE), 2019.

[21] H. Li and Y. Li. "LogSpy: System Log Anomaly Detection for Distributed Systems," 2020 International Conference on Artificial Intelligence and Computer Engineering (ICAICE), 2020.

[22] F.-J. Yang, "An Extended Idea about Decision Trees," 2019 International Conference on Computational Science and Computational Intelligence (CSCI), Las Vegas, NV, USA, 2019, pp. 349-354.

[23] Natekin, Alexey & Knoll, Alois, "Gradient Boosting Machines, A Tutorial." Frontiers in neurorobotics, 2013.