PROCEEDINGS OF SCIENCE

Sentence-BERT and Transformer-Based Log Anomaly Detection

Yuanyuan Liu,[a,b,*] **Jiarong Wang**⊠**,**[a] **Tian Yan**[a] **and Fazhi Qi**[a,c]

[a] *Computing Center, Institute of High Energy Physics, Chinese Academy of Sciences,*
Beijing 100049, P.R.China

[b] *School of Nuclear Science and Technology, University of Chinese Academy of Sciences,*
Beijing 100049, P.R.China

[c] *Spallation Neutron Source Science Center,*
Dongguan 523803, P.R.China

E-mail: wangjr@ihep.ac.cn

The increasing scale of large-scale scientific facilities and scientific data center network systems makes the system vulnerable to internal failures or external attacks during operation, leading to system paralysis and significantly impacting the normal work and life of users. Timely and accurate fault location is crucial for ensuring stable system operation. Log data, which records the system's running status, plays a vital role in identifying and rectifying anomalies that occur during system operation. Therefore, timely troubleshooting of log anomalies is essential for maintaining the stability of complex systems and ensuring the safe operation of large scientific facilities and scientific data centers.

This paper proposes a log anomaly detection method based on local information extraction in a Transformer model, which learns the deep language features and context information of log recording through a Sentence-BERT model. The local feature extraction of convolution operation and the Transformer model are used to capture context information in the sequence to improve the model's ability to recognize complex patterns and speed up model training and reasoning. We have carried out experiments on log data sets, and the experimental results show that this method can provide a reliable and efficient solution for log anomaly detection of large scientific research facilities and scientific data center network systems.

International Symposium on Grids and Clouds (ISGC2024)
24 -29 March, 2024
Academia Sinica Computing Centre (ASGC), Institute of Physics, Academia Sinica Taipei, Taiwan

[*] Speaker

1. Introduction

System logs are important data sources that record information such as system running status, user operations, and abnormal events. They are an indispensable part of understanding system behaviors and ensuring system stability and security.[1] Anomaly detection of system logs can help locate anomalies and analyze faults, which is a topic worth studying at present.

Log anomaly detection aims to identify abnormal behaviors that are significantly different from normal behavior patterns by analyzing data patterns and behavior characteristics in system logs, so as to realize early warning of potential threats and system failures. This not only requires a deep understanding of the structure and meaning of various system logs, but also requires the use of advanced technical means such as data mining and machine learning to improve the accuracy and efficiency of anomaly detection.

With the development of information technology, the scale of modern data center systems is expanding day by day, and the log records generated by complex systems have reached tens of millions. Even a system deployed on a medium-scale network generates more than terabytes of logs per day.[2] Research on log anomaly detection mainly focuses on anomaly detection methods based on machine learning and anomaly detection algorithms based on deep learning. Machine learning-based algorithms can automatically adapt to changes in log data, but often require large amounts of labeled data and are not effective when dealing with extremely unbalanced data and unlearned anomalies.

At present, the research on log anomaly detection mainly focuses on log anomaly detection algorithms based on deep learning, but the existing methods also have some defects. As unstructured data, the output format of logs is highly unstable, and the log format varies significantly between different systems. Therefore, the parsing accuracy of logs will affect the effect of anomaly detection model.[3]In addition, the deep learning model has the characteristics of large number of parameters and long training time.

Log anomaly detection is not only an important research direction in the field of network security, but also a basic work to ensure the stable and reliable operation of information system. To develop a more efficient and intelligent log anomaly detection model is very important to improve the efficiency of operation and maintenance personnel and the reliability of large-scale systems.[4] In order to improve the accuracy of log analysis and the efficiency of log detection, this paper proposes a log anomaly detection algorithm based on the Transformer model based on local information extraction.

The contributions of this paper are as follows:

(1) A simpler log preprocessing method is proposed, which does not need to parse the log template, retains the detailed semantic information of the log, and avoids the influence of the existing log parser on the inaccuracy of log parsing;

(2) Extract the full semantic features of log statements, and use the pre-trained Sentence-BERT model to fully consider the impact of words in log statements on context during semantic

embedding;

(3) The anomaly detection model is constructed, and the convolution layer and Transformer model are combined to extract local features and global features, and the parallel processing capability of tranformer model is utilized to improve detection efficiency. Through the evaluation of system log data sets generated in two different systems, it is proved that our method can effectively complete the log anomaly detection task.

The rest of the paper is organized as follows. Section 2 introduces related work and techniques. Section 3 describes the system architecture and the implementation of the algorithm. Section 4 presents some experiments of comparison between the proposed method and other log anomaly detection algorithms. Section 5 provides conclusions.

2. Related Works

Researchers at home and abroad have proposed many methods in log anomaly detection and achieved many important research results. The early traditional log anomaly detection method [5][6] relies on the operation and maintenance personnel to use regular expression matching to manually analyze the log. With the expansion of the system scale, the log data explosion increases, and the efficiency and accuracy of the traditional log anomaly detection method are difficult to meet the needs. At present, most researchers apply machine learning methods and deep learning methods to log anomaly detection, which greatly improves the accuracy of log anomaly detection.

2.1 The machine learning based anomaly detection methods

Machine learning log anomaly detection methods can be divided into supervised machine learning log anomaly detection methods and unsupervised log anomaly detection algorithms[7]. The supervised learning method uses labeled data sets, and because there are fewer labeled data sets in the actual environment, the unsupervised learning method is widely used because of the advantage of using unlabeled data sets during training.

Supervised log anomaly detection algorithms such as Han et al.[8]implement SVM-based log anomaly detection algorithms. Experts merge semantic similar log templates, and use online learning theory to dynamically update parameters. Finally, SVM-based log anomaly detection is performed.

Unsupervised log anomaly detection such as the improved isolated forest(IF) algorithm proposed by Xu et al.[9] provides an efficient anomaly detection mechanism for structured continuous data. The core principle of this method is that normal data instances tend to form high-density regions in the data space, while abnormal instances are sparsely distributed outside these high-density regions. If the instances of log data deviate significantly from the space composed of normal data instances, these instances can be identified as exceptions. However, the study also points out that when dealing with large-scale data sets, the performance of the isolated forest algorithm in detecting outliers may be negatively affected.

2.2 The deep learning based anomaly detection methods

Log data may contain a variety of types of information, such as timestamp, log level, message text, etc. In the face of large-scale data sets, deep learning models can process these different dimensions of information through different layers of the network. Deep learning models, especially Recurrent Neural Network[10], Long Short-Term Memory[11] and Gated Recurrent Unit[12], are designed to process such data and can capture long-term dependencies, which is very important for identifying abnormal patterns of logs.

LSTM is a variant of recurrent neural networks, which has attracted attention because of its ability to maintain long-term dependence on information when processing sequence data. This ability makes LSTM very suitable for log anomaly detection tasks. DeepLog proposed by Du et al.[13] is a classical log anomaly detection algorithm based on LSTM. It detects abnormal behaviors in log template sequences and parameter values by stacking two layers of LSTM networks.Meng et al.[14] proposed LogAnomaly.By introducing dLCE (a natural language processing model), word embedding technology is used to semantically express log templates, and then template vectors are generated. These vectors are then used to train the LSTM model to predict and match the newly generated log templates during system operation, and anomaly detection is performed based on the similarity between template vectors. At the same time, the LogNL proposed by Zhu et al.[15] combines the advantages of DeepLog and LogAnomaly, and uses a two-layer LSTM network to perform abnormal judgment on log templates and parameter values, which further improves the performance of the model.In addition, Hashemi et al.[16] proposed a siamese network model based on LSTM.This model maps log sequences to vector space, so that sequences of the same type are close to each other in vector space, while maximizing the distance between different types of sequences, thereby improving the accuracy of anomaly detection.

There are still some limitations when using LSTM or GRU to deal with log anomaly detection tasks, which is mainly reflected in the inefficiency of dealing with large-scale data sets.[17] Compared to RNN, Transformer[18] can process the entire sequence in parallel. This not only improves efficiency, but also makes it have better performance and faster training speed than RNN-based models when dealing with long sequences. Huang et al.[19]proposed the HitAnomaly algorithm for the first time using the Transformer model in the log anomaly detection task. They converted the log template and parameter values into vector representations, and then used the attention mechanism to merge. Finally, the abnormal probability distribution is obtained through the Softmax layer. Guo et al.[20] proposed a framework called TransLog, which aims to improve the generalization ability of the model on multi-domain log data, and enhance the generalization and adaptability of the model through pre-training and Transformer-based adjustment stages.

3. System Architecture

The log anomaly detection method proposed in this paper consists of four modules, which are log preprocessing module, log sequence generation module, semantic embedding module and log anomaly detection module. The overall architecture is shown as follows:

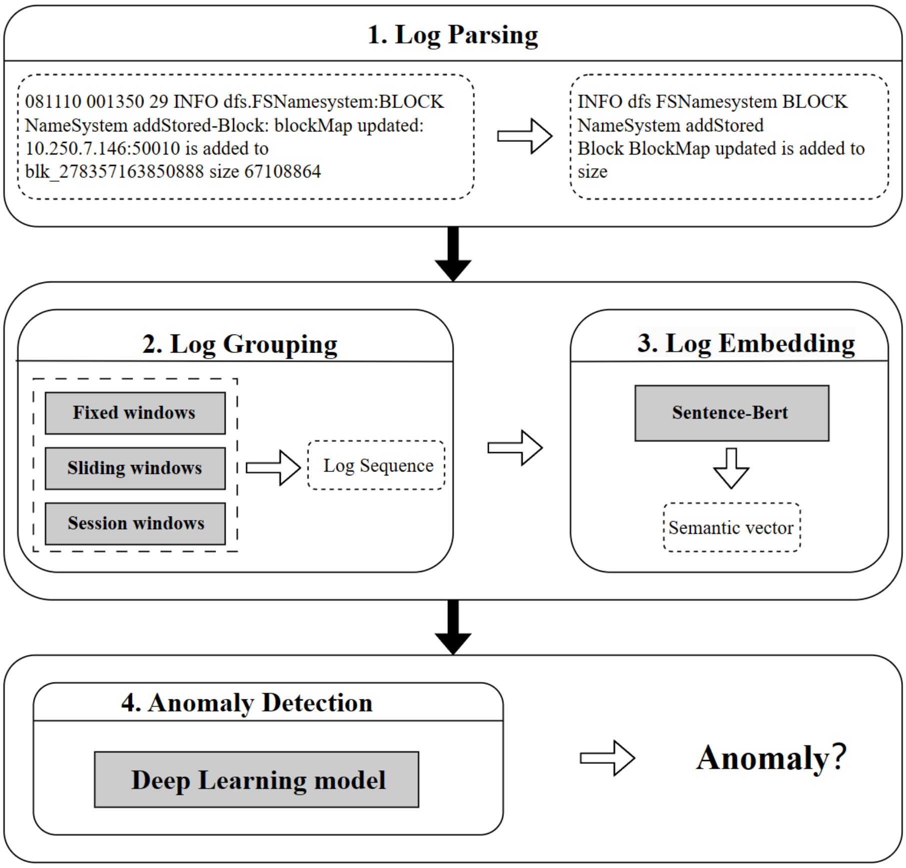
1. Log Parsing
081110 001350 29 INFO dfs.FSNamesystem:BLOCK NameSystem addStored-Block: blockMap updated: 10.250.7.146:50010 is added to blk_278357163850888 size 67108864
INFO dfs FSNamesystem BLOCK NameSystem addStored Block BlockMap updated is added to size
2. Log Grouping
Fixed windows
Sliding windows
Session windows
Log Sequence
3. Log Embedding
Sentence-Bert
Semantic vector
4. Anomaly Detection
Deep Learning model
Anomaly?

Figure 1: System architecture.

3.1 Log Preprocessing

Different systems may have their own unique log format and log content, but the components of a log can be divided into log header and log body. Among them, the log header contains timestamp, log level, source, etc. and the log body contains the specific content of the log. Log preprocessing needs to extract effective information from semi-structured log records. Most of the existing log anomaly detection methods are divided into two parts : (1) Using the log parser to parse the log statement to obtain the log template[21] ; (2) Anomaly detection of log sequences using machine learning or deep learning techniques. However, the method of preprocessing with log parser may bring noise influence and affect the performance of model detection. There are also cases where there is a deviation in semantic understanding and the parameter variables are parsed into templates.

Considering the problems existing in the log parsing method, this paper only performs simple preprocessing on the log statement and retains the semantic information of the log statement. Firstly, delete the non-letter content, including timestamp, IP address, PID, etc., and then replace the comma, underline and other characters as spaces. Avoid affecting log anomaly detection performance due to low log parsing accuracy and log semantic loss.

Table 1: Log parsing example

Logs	Interface ae3, changed state to down
	Interface ae3, changed state to up
Error template	Interface <*>, changed state to <*>
Correct template	Interface <*>, changed state to down
	Interface <*>, changed state to up

3.2 Log Sequence Generation

The events or states that occur in the system may be described by more than one log statement, and the relationship between contexts needs to be considered. Therefore, the system log needs to divide the log sequence according to a certain window size. There are two main ways to divide the log sequence : according to the session ID division and according to the window division.[22]

According to the session ID division : This division method is mainly based on the log statement contains the relevant session identifier. For example, in the HDFS dataset, each log contains a block id identifier, and log statements containing the same block id can be divided into the same log sequence according to the block id identifier.

According to the window division : this division method is divided into fixed window division, sliding window division, etc. For a fixed window, you need to set the window size value that is the length of the log sequence, and there is no overlap between the windows, a log will only appear in a log sequence. For the sliding window, it is necessary to set the window size value and the step size value. In general, the step size value is less than the window size value. The sliding window slides with the step size value each time, and there is overlap between the windows. A log may appear in multiple log sequences.

The data sets used in the subsequent experiments in this paper are data sets with session identification and data sets without session identification. Therefore, in this module, the log data set with session identification is divided according to the session ID, and the data set without session identification is divided according to the window, and a more flexible sliding window is used to divide the log sequence.

3.3 Semantic Embedding

Sentence-BERT[23] is a derivative model of BERT[24], which is specifically used to generate sentence-level embedding vectors. Although the BERT model performs well on many NLP tasks, it is less efficient when it is directly used for tasks such as sentence similarity calculation, because it needs to compare each pair of sentences in pairs, which may be very time-consuming in practical applications. The Sentence-BERT model is based on the twin network and the triplet network. Through Sentence-BERT, sentences can be quickly converted into sentence vectors, and rich semantic information can be retained. The structure is shown in Figure 2.

The preprocessed log statement is only composed of words, which can be regarded as a statement with special meaning. At present, semantic embedding for log statements requires a

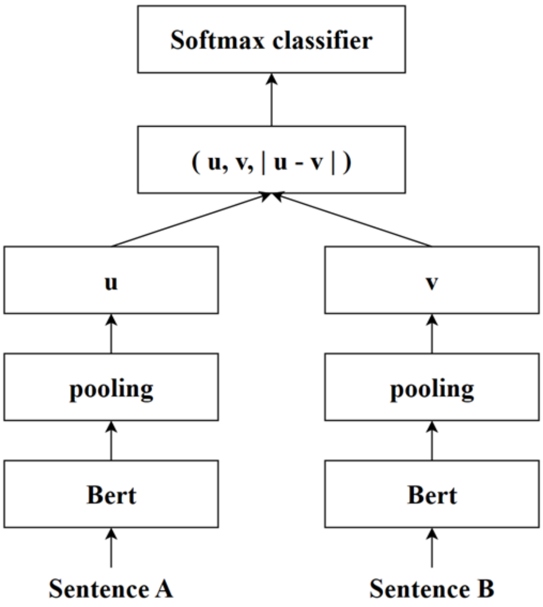

Figure 2: Sentence-BERT

large number of events and computing resources for model training. This paper chooses to use the Paraphrase-MiniLM-L6-v2 model[25]. Paraphrase-MiniLM-L6-v2 is a model based on MiniLM optimization, which is used to generate sentence embedding. The Sentence-BERT framework is specially fine-tuned to generate excellent sentence embedding to optimize the performance of specific NLP tasks.

3.4 Log Anomaly Detection

The Transformer model is a popular architecture in natural language processing. It processes sequence data through a self-attention mechanism, and has obvious advantages in processing long sequences and capturing global dependencies. In some cases, the Transformer model may rely too much on global information and ignore the capture of local key signals. The convolution operation in the convolutional neural network has the characteristics of extracting local information. Therefore, by combining the advantages of the two, this paper proposes a Transformer model based on local information extraction to make up for the limitations of the Transformer model to capture local features, so as to achieve the best results in the log anomaly detection task.

The log sequence is in the form of one-dimensional data, so this paper uses one-dimensional convolution Conv1D for processing. However, the limited receptive field of a single convolution operation limits its ability to capture wider context information. In order to overcome this limitation and enhance the performance of the model in local information mining, this paper proposes a multi-layer convolution operation. Multiple convolution layers are stacked sequentially, and each layer is equipped with convolution kernels of different sizes. This multi-layer convolution operation aims to capture the local features of log sequence data at different abstract levels, and finally extract the features of the information calculated by different convolution kernels through maximum pooling.

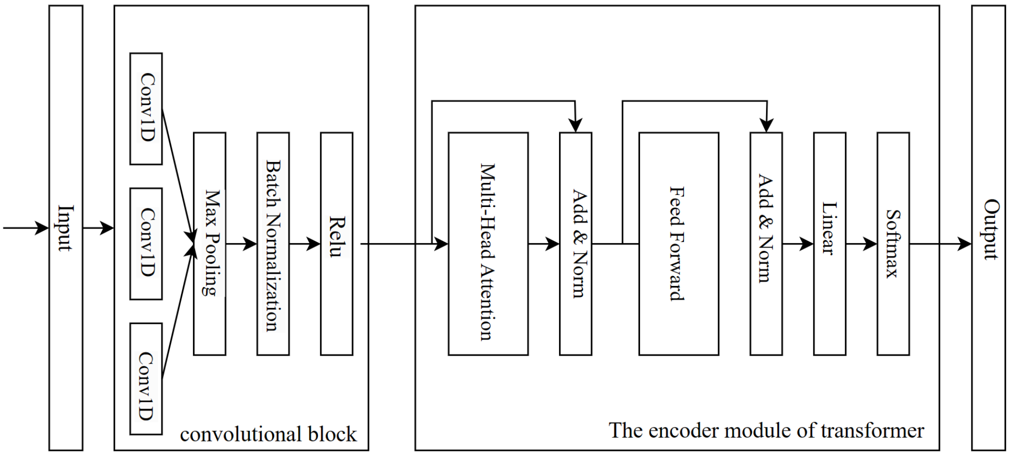

Figure 3: Log Anomaly Detection Module

In order to improve the accuracy of log anomaly detection, we combine the convolution operation with the Transformer model to form a Transformer model for local information extraction. Log data is processed by log preprocessing module, log sequence generation module, semantic embedding module and log anomaly detection module, and finally the probability of normal or abnormal is obtained.

4. Experiment

4.1 Dataset

We use two data sets collected from real application environments : HDFS(Hadoop Distributed File System) and BGL(Blue Gene/L Supercomputer). The specific information is shown in Table. Both data sets are obtained from the large log data set LogHub of the Intelligent Operation and Maintenance Research Team of the Chinese University of Hong Kong[26].

Table 2: Dataset information

DataSet	HDFS	BGL
Time Span	38.7 hours	214.7 days
Raw Size	1.47 GB	708.76 MB
Number of logs	11175629 (logs)	4747963 (logs)
Number of abnormal logs	16838 (blocks)	348460 (logs)

The HDFS data set is generated by the HDFS cluster of 203 nodes. During the log generation process, exceptions are manually identified and marked according to certain rules. Each log record associated with block id is marked with a normal or abnormal label, and the log sequence is segmented according to block id.

The BGL dataset was collected by the Blue Gene/L Supercomputer system of Lawrence Livermore National Laboratory (LLNL). The log contains abnormal and non-abnormal logs identified by the ' - ' category tag. In the first column of the log, ' - ' denotes a non-anomaly log, while the other logs are exception logs[27].

4.2 Experimental Setups

The experimental development language is Python3.8, using the deep learning framework Tensorflow 2.4.0 to build the framework, running on the Ubuntu20.04 operating system, and using GeForce RTX 3090 GPU to accelerate model training.

Log anomaly detection can be regarded as a binary classification problem, so the experiment uses Precision, Recall and F1-score as evaluation indicators. Precision refers to the proportion of logs that are actually abnormal in all logs that are identified as abnormal. The recall rate refers to the proportion of logs that are correctly identified as abnormal in all logs that are actually abnormal ; f1-score is the harmonic mean of precision and recall, which is used to balance the performance of precision and recall. The specific calculation method is as follows :

$$Precision = \frac{TP}{TP + FP}$$

$$Recall = \frac{TP}{TP + FN}$$

$$F1 - score = \frac{2 \cdot Precision \cdot Recall}{Precision + Recall}$$

4.3 Experimental Results

In order to verify the effect of this method on the log anomaly detection task, different window sizes and step sizes are selected on the BGL dataset for experiments. The BGL dataset is divided into 8 : 2, the sliding window sizes are 20 and 10, and the step size is half of the window size. From the experimental results, it can be seen that the anomaly detection effect of the model under different sliding windows is stable, indicating that the method in this paper has certain robustness.

Table 3: Experimental Results

Window size	Step size	Precision	Recall	F1-score
20	10	0.99	0.98	0.99
10	5	0.98	0.97	0.98

4.4 Ablation Experiment

In this section, the Transformer model of local information extraction is applied to the task of log sequence anomaly detection, and its effectiveness is verified and analyzed by experiments. The

HDFS data set is divided into 7 : 3. In the part of log anomaly detection, the convolution layer of local information extraction and the original Transfromer model are used as the basic model, and compared with the Transformer model of local information extraction proposed in this paper.

Table 4: Ablation Experiment

	Precision	Recall	F1-score
Conv1D	0.91	1.00	0.95
Transformer	0.81	0.98	0.87
Our Method	0.95	1.00	0.97

The table shows the experimental results of the model on the HDFS data set. When the anomaly detection model is a convolutional layer, it can effectively extract local information in the log sequence. This is also due to the fact that the Sentence-BERT model based on sentence vector can learn rich and complex semantic information, and can well capture the complex relationship and semantic meaning between texts. When the anomaly detection model is the Transformer model, it processes the sequence data through the self-attention mechanism. The Transformer model has obvious advantages in processing long sequences and capturing global dependencies, but it has some potential limitations in capturing fine-grained local information.

Considering the advantages of the convolutional layer and the Transformer model, we propose a Transformer model for local information extraction, which combines the local feature recognition ability of the one-dimensional convolutional layer and the global context understanding ability of the Transformer model. The experimental results show that our method is effective for log anomaly detection tasks.

4.5 Comparative experiment

Furthermore, to compare the log anomaly detection performance of our proposed model, we conducted comparative experiments. Zhang et al. proposed LogRobust[28], an attention-based Bi-LSTM model framework for log anomaly detection. LogRobust transforms each log statement within a log sequence into a semantic vector, utilizing the proposed anomaly detection model to perform log anomaly detection. This model is capable of capturing the contextual information within log sequences and automatically learning the importance of different logs. LogRobust is designed to handle noise and dynamic changes in log data, thereby enhancing the robustness of anomaly detection.

Table 5: Comparative experiment

	Precision	Recall	F1-score
LogRobust	0.88	0.94	0.91
Our Method	0.98	0.97	0.98

In our research, we conducted experiments to compare our methodology with the LogRobust approach using the publicly available BGL dataset. The dataset was automatically divided into a training set and a test set in an 8:2 ratio, utilizing a window size of 10 and a step of 5. The results are presented in the table. Our method achieved a Precision of 0.98, while LogRobust recorded a Precision of 0.88, indicating some false positives with the LogRobust approach where it incorrectly classified certain normal log sequences as anomalies. The experimentation demonstrates the superior performance of our method in log anomaly detection tasks, showing higher Precision, Recall, and F1-score. Our approach exhibits better performance in reducing false positives and is more effective in identifying true anomalies.

5. Conclusion

This paper proposes a Transformer model based on local information extraction, capable of simultaneously learning the log's local and global features. It employs simple preprocessing during the log parsing phase and uses a pre-trained Sentence-BERT model for semantic embedding. Compared to traditional log parsers and word vector models, this method reduces the impact of noise and learns richer, more complex semantic information, while requiring fewer parameters and computational resources. Experimental results demonstrate that our method performs well in log anomaly detection tasks.

Log anomaly detection, as a crucial aspect of intelligent operation and maintenance, requires not only the rapid and accurate detection of anomalies but also focuses on understanding the causes of anomalies and their relationships. This is also the direction we need to explore in the future research.

Acknowledgments

This work is supported in part by the National Key Research and Development Program of China under grant no.2023YFC2605703 and the specialized project for cybersecurity and informatization in the 14th Five-Year Plan of CAS under grant no.WX145XQ12 and Chinese Academy of Sciences Special Project on Science and Technology Platform: National High Energy Physics Science Data Center.

References

[1] Jiarong Wang, Tian Yan, Dehai An, Zhongtian Liang, Chaoqi Guo, Hao Hu, Qi Luo, Hongtao Li, Han Wang, Shan Zeng, et al. A comprehensive security operation center based on big data analytics and threat intelligence. In *International Symposium on Grids & Clouds*, pages 22–26, 2021.

[2] Yali Yuan, Sripriya Srikant Adhatarao, Mingkai Lin, Yachao Yuan, Zheli Liu, and Xiaoming Fu. Ada: Adaptive deep log anomaly detector. In *IEEE INFOCOM 2020-IEEE Conference on Computer Communications*, pages 2449–2458. IEEE, 2020.

[3] Pinjia He, Jieming Zhu, Shilin He, Jian Li, and Michael R Lyu. An evaluation study on log parsing and its use in log mining. In *2016 46th annual IEEE/IFIP international conference on dependable systems and networks (DSN)*, pages 654–661. IEEE, 2016.

[4] Tian Yan, Hao Hu, Dehai An, Fazhi Qi, and Chen Jiang. Cyber security monitoring and data analysis at ihep. In *International Symposium on Grids & Clouds 2019*, page 11, 2019.

[5] James E Prewett. Analyzing cluster log files using logsurfer. In *Proceedings of the 4th Annual Conference on Linux Clusters*, pages 1–12. Citeseer State College, PA, USA, 2003.

[6] John P Rouillard. Real-time log file analysis using the simple event correlator (sec). In *LISA*, volume 4, pages 133–150, 2004.

[7] D Ajith. A survey on anomaly detection methods for system log data. *International Journal of Science and Research (IJSR)*, 2019.

[8] Shangbin Han, Qianhong Wu, Han Zhang, Bo Qin, Jiankun Hu, Xingang Shi, Linfeng Liu, and Xia Yin. Log-based anomaly detection with robust feature extraction and online learning. *IEEE Transactions on Information Forensics and Security*, 2021.

[9] Dong Xu, Yanjun Wang, Yulong Meng, and Ziying Zhang. An improved data anomaly detection method based on isolation forest. In *2017 10th international symposium on computational intelligence and design (ISCID)*, volume 2, pages 287–291. IEEE, 2017.

[10] Jeffrey L. Elman. Finding structure in time. *Cognitive Science*, 14(2):179–211, 1990.

[11] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 1997.

[12] Junyoung Chung, Çaglar Gülçehre, KyungHyun Cho, and Yoshua Bengio. Empirical evaluation of gated recurrent neural networks on sequence modeling. *CoRR*, abs/1412.3555, 2014.

[13] Min Du, Feifei Li, Guineng Zheng, and Vivek Srikumar. Deeplog: Anomaly detection and diagnosis from system logs through deep learning. In *Proceedings of the 2017 ACM SIGSAC conference on computer and communications security*, pages 1285–1298, 2017.

[14] Weibin Meng, Ying Liu, Yichen Zhu, Shenglin Zhang, Dan Pei, Yuqing Liu, Yihao Chen, Ruizhi Zhang, Shimin Tao, Pei Sun, et al. Loganomaly: Unsupervised detection of sequential and quantitative anomalies in unstructured logs. In *IJCAI*, volume 19, pages 4739–4745, 2019.

[15] Bei Zhu, Jing Li, Rongbin Gu, and Liang Wang. An approach to cloud platform log anomaly detection based on natural language processing and lstm. In *Proceedings of the 2020 3rd International Conference on Algorithms, Computing and Artificial Intelligence*, pages 1–7, 2020.

[16] Shayan Hashemi and Mika Mäntylä. Detecting anomalies in software execution logs with siamese network. *arXiv preprint arXiv:2102.01452*, 2021.

[17] Yinglung Liang, Yanyong Zhang, Hui Xiong, and Ramendra Sahoo. Failure prediction in ibm bluegene/l event logs. In *Seventh IEEE International Conference on Data Mining (ICDM 2007)*, pages 583–588. IEEE, 2007.

[18] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 2017.

[19] Shaohan Huang, Yi Liu, Carol Fung, Rong He, Yining Zhao, Hailong Yang, and Zhongzhi Luan. Hitanomaly: Hierarchical transformers for anomaly detection in system log. *IEEE transactions on network and service management*, 2020.

[20] Hongcheng Guo, Xingyu Lin, Jian Yang, Yi Zhuang, Jiaqi Bai, Tieqiao Zheng, Bo Zhang, and Zhoujun Li. Translog: A unified transformer-based framework for log anomaly detection. *arXiv preprint arXiv:2201.00016*, 2021.

[21] Jieming Zhu, 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)*, pages 121–130. IEEE, 2019.

[22] Shilin He, Jieming Zhu, Pinjia He, and Michael R Lyu. Experience report: System log analysis for anomaly detection. In *2016 IEEE 27th international symposium on software reliability engineering (ISSRE)*, pages 207–218. IEEE, 2016.

[23] Nils Reimers and Iryna Gurevych. Sentence-bert: Sentence embeddings using siamese bert-networks. *arXiv preprint arXiv:1908.10084*, 2019.

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

[25] Wenhui Wang, Furu Wei, Li Dong, Hangbo Bao, Nan Yang, and Ming Zhou. Minilm: Deep self-attention distillation for task-agnostic compression of pre-trained transformers. *CoRR*, abs/2002.10957, 2020.

[26] Jieming Zhu, Shilin He, Pinjia He, Jinyang Liu, and Michael R Lyu. Loghub: A large collection of system log datasets for ai-driven log analytics. In *2023 IEEE 34th International Symposium on Software Reliability Engineering (ISSRE)*, pages 355–366. IEEE, 2023.

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

[28] Xu Zhang, Yong Xu, Qingwei Lin, Bo Qiao, Hongyu Zhang, Yingnong Dang, Chunyu Xie, Xinsheng Yang, Qian Cheng, Ze Li, et al. Robust log-based anomaly detection on unstable log data. In *Proceedings of the 2019 27th ACM joint meeting on European software engineering conference and symposium on the foundations of software engineering*, pages 807–817, 2019.