ISBN: 978-981-94-4198-3
2025 the 15th International Workshop on Computer Science and Engineering (WCSE 2025)
doi: 10.18178/wcse.2025.06.036

Relevance-Based Data Selection for BERT-Based Anomaly Detection Using Unstructured Logs

Wei-Ting Chang [1], Ren-Hung Hwang [1] + and Jian-Liang Pan [1]

[1]National Yang Ming Chiao Tung University, Tainan, Taiwan

Abstract. As software systems grow in scale and complexity, log-based automated anomaly detection has become an essential tool for maintaining system reliability. However, machine learning and deep learning-based detection methods typically require pre-labeled data for training, which poses a challenge due to the vast volume and repetitive nature of logs generated by large systems. Furthermore, log formats often evolve with system updates, making traditional log parsers prone to errors that can negatively affect anomaly detection performance. To address these challenges, this study proposes a robust anomaly detection system that directly processes unstructured logs. The system employs a BERT tokenizer for tokenization and utilizes relevance-based selection and clustering techniques to extract less than 0.01% of high-quality training data from millions of unlabeled logs. Additionally, BERT is leveraged to capture both sequential and semantic information in the logs, facilitating the automated detection of normal and anomalous patterns. Experimental results demonstrate that the proposed method achieves an F1-score exceeding 0.96 across four supercomputer datasets, and an F1-score above 0.91 for the detection of previously unseen events.

Keywords: anomaly detection, data selection, relevance, unstructured logs

1. Introduction

A software system comprises interdependent components and services, which are monitored through system logs that record execution status, loaded components, and runtime errors. As system complexity increases, traditional manual log analysis becomes inadequate due to the growing volume and complexity of log data. To address this, automated anomaly detection systems, often utilizing machine learning or deep learning techniques, have been developed to efficiently analyze logs and identify potential anomalies by detecting deviations from normal patterns [1].

Previous research has explored various approaches to develop log-based anomaly detection systems, ranging from traditional data mining techniques [2][3] to advanced deep learning methods [4][5]. The typical implementation process of a log-based anomaly detection system [6], as illustrated in Fig. 1, generally involves the use of log parsers to convert raw logs into specific templates before encoding and model training [7][8].

Despite the strong performance demonstrated in previous studies, several challenges persist. First, most existing approaches rely on supervised or semi-supervised models, which require a significant amount of labeled data. However, small and medium-sized software systems can generate hundreds to thousands of logs per minute, while large systems, such as cloud platforms, may produce millions of logs per minute. Managing such vast amounts of data and performing manual labeling is both time-consuming and labor-intensive. Moreover, log data often contains substantial repetition, which can reduce the quality of the training data and subsequently degrade model performance [9][10]. Second, many studies depend on log parsing to convert logs into predefined templates for further processing. However, log formats frequently change due to system updates, leading to potential parsing errors or detection inaccuracies [11][12]. Therefore, there is a critical need for more flexible and adaptive methods to handle these changes in log structure, ensuring both parsing accuracy and detection performance.

To tackle the aforementioned challenges, this study proposes an efficient and highly robust anomaly detection system. First, the system eliminates the need to convert logs into predefined templates, instead directly processing unstructured logs using a BERT tokenizer based on the WordPiece algorithm for

+ Corresponding author. Tel.: +8863032121 #57729; fax: +8863032121 #57729.
E-mail address: rhhwang@nycu.edu.tw.

tokenization. Second, it leverages unlabeled data and employs relevance-based data selection, significantly reducing processing complexity and training costs while ensuring the provision of high-quality training data for the detection model. Finally, the system utilizes BERT to capture both semantic and sequential information from the logs, allowing it to identify normal and anomalous patterns and enabling automated detection of system anomalies. The main contributions of this study are as follows:

- We propose a relevance-based automated data selection method capable of identifying and extracting critical data from large volumes of unlabeled, unstructured logs for training purposes.
- We optimize the preprocessing method by removing the need for log parsing, allowing the detection model to more effectively capture the sequential and semantic information within the logs.

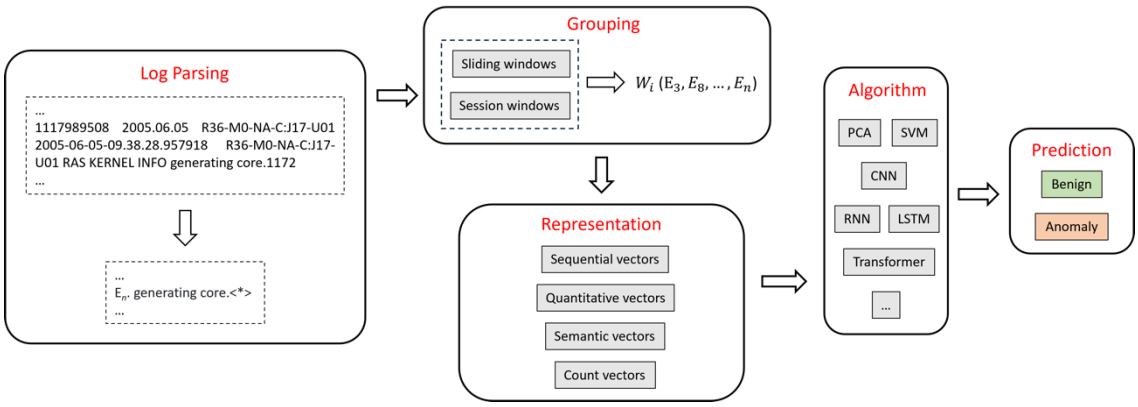


Fig. 1: Workflow of log-based anomaly detection system.

The remainder of this paper is organized as follows: Section II reviews related work in the area of log-based anomaly detection. Section III defines the problem this research addresses. Section IV presents the proposed methodology, followed by the performance evaluation in Section V. Finally, Section VI concludes the paper.

2. Related Work

The related works are summarized in Table 1, with most approaches requiring log parsing to extract keywords and convert them into event templates.

DeepLog [4] introduces a method that parses logs to extract keyword and parameter value vectors. It then uses an LSTM to predict the next log entry. If the actual log entry does not match the prediction, it is classified as an anomaly. LogBERT [13] employs a technique called masked log key prediction (MLKP), which predicts the probability of a log key appearing at a specific position in a log sequence. If the actual log key does not match the predicted set of possible keys, it is marked as an anomaly. LogAnomaly [5] presents the Template2vec method, which leverages labeled synonym and antonym templates to identify events that the model has not previously encountered. PLELog [14] begins with a portion of labeled normal data and clusters the remaining unlabeled data using this initial set. Based on the clustering results, the unlabeled data are classified as either normal or abnormal, addressing the challenge of insufficient labeled data. LogBD [15] parses the logs, encodes them using BERT, and classifies them with a temporal convolution network (TCN) with adversarial training.

In addition to these methods, several other approaches have also been developed to improve log-based anomaly detection by focusing on encoding techniques and addressing specific challenges such as log parsing and out-of-vocabulary (OOV) issues. LogRobust [12] encodes parsed log events using FastText and applies weights calculated by TF-IDF to capture both the semantics and the significance of the events. HilBERT [16] parses the logs and tokenizes them using a BERT tokenizer. A single log sequence is first input into BERT to compute embeddings, after which multiple embedded sequences are fed into a classifier based on the attention mechanism. NeuralLog [11] does not rely on log parsing. Instead, it filters out numbers and punctuation from the logs, tokenizes them using the WordPiece algorithm, and calculates embeddings with BERT to address the out-of-vocabulary (OOV) problem commonly encountered in logs.

While many of the aforementioned models rely on LSTM, Transformer, or other sequence-based architectures, traditional deep learning techniques have also been effectively utilized for anomaly detection. For example, LogCNN [17] vectorizes log keys and uses them as input data to train a Convolutional Neural Network (CNN) model.

Given the large volume of log data, previous methods have often selected a small subset of data based on chronological order for training purposes. However, this approach can compromise dataset quality due to the limitations imposed by the selected time frame. Additionally, log data frequently exhibits a high degree of redundancy, which can result in an incomplete dataset. This poses a significant challenge when building anomaly detection models, as determining the optimal amount of data for effective training becomes difficult.

To overcome these challenges, we propose a method for selecting critical data from a large, unlabeled dataset. By eliminating the need for log parsing, our approach mitigates log instability, thereby enhancing the robustness of the detection system.

Table 1: Related work

Paper	Initial data selection	Template	Training Strategy	Representation	Algorithm
[4]	Chronological (Benign only)	V	Semi-supervised	Log key	LSTM
[13]				Log key	MLKP+VHM
[5]				Template2vec	LSTM
[14]				FastText	Attention-based GRU
[15]				BERT	TCN + Adversarial training
[11]	Chronological	X	Supervised	BERT	Attention-based classifier
[16]		V		Sequential BERT	Attention-based classifier
[17]				Logkey2vec	CNN
[12]	Down sampling to balanced data			FastText + TF-IDF	Attention-based Bi-LSTM
Ours	Sentence BERT + HDBSCAN	X	Supervised	Special token + BERT	BERT-based classifier

3. Problem Definition

To efficiently process large volumes of log data and develop a robust log-based anomaly detection system, we propose a method that provides a strategy for selecting a small amount of high-quality training data from massive log datasets. This approach reduces processing complexity, lowers training costs, and strengthens the detection system's robustness. By utilizing unstructured log data directly, without the need for parsing or template conversion, the method first selects data based on relevance. Subsequently, it only requires labelling and training the model on a small amount of data. The problem is defined as follows:

- Input: The dataset comprises large-scale, unstructured logs containing various forms of information, including timestamps, event descriptions, and potential error messages.

- Output: The desired output is the classification of the test data into one of two categories: benign or anomalous.
- Objectives: (1) To improve the model's performance in accurately classifying the data. (2) To ensure the broad applicability of the selected dataset for future research studies.

4. Methodology

4.1. Overview

Fig. 2 illustrates the framework of our system, which consists of three key components: data preprocessing, data selection, and model training. The framework encompasses both the offline model training phase and the online anomaly detection process.

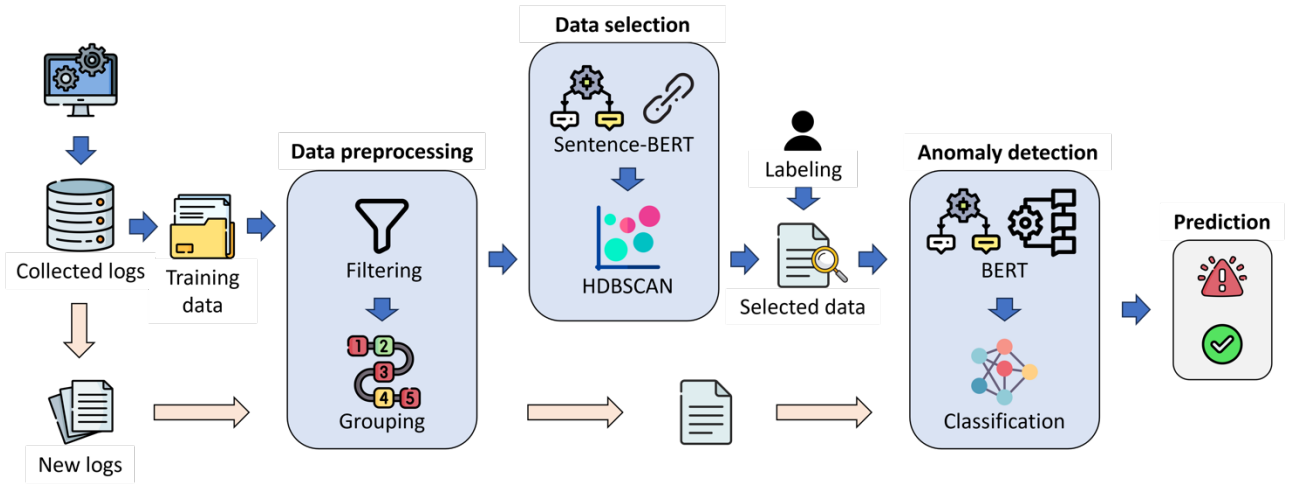


Fig. 2: Framework of the proposed system.

During the offline model training phase, the collected unlabeled log data is first preprocessed through filtering and grouping. Next, the data selection process is applied, and the selected data are labeled and fed into the detection model for supervised learning. In the online anomaly detection phase, the preprocessed log data is directly input into the trained detection model for real-time anomaly detection. If classification errors occur during online usage, the erroneous data can be collected and later used to fine-tune the detection model.

4.2. Data Preprocessing

At this stage, the collected log data is preprocessed to serve as input for subsequent methods. Referencing the filtering approach proposed in NeuralLog, this involves using regular expressions to remove numbers and punctuation marks from the logs, eliminating elements that do not significantly contribute to semantic understanding. Unlike traditional log parsers, this method does not rely on predefined templates, making it more suitable for handling irregular or unstructured log entries and ensuring robustness against changes in log recording formats.

However, certain combinations of numbers and punctuation, such as IP addresses, carry specific meanings. Simply filtering them out could result in information loss, potentially impacting the model's ability to recognize important sequences. To prevent this, we replace these symbols with special tokens from BERT, allowing the model to better capture the contextual meaning, as shown in Fig. 3. This approach enhances the processing of log data without sacrificing critical information.

4.3. Data Selection

To effectively leverage unlabeled data and select important entries, this phase employs a relevance-based method for unsupervised data selection.

At this stage, we use Sentence-BERT (SBERT) [18] to compute embeddings for each sequence, followed by the application of the Hierarchical Density-Based Spatial Clustering of Applications with Noise (HDBSCAN) algorithm [19] for clustering. After clustering, we select the top n points with the highest density from each cluster for training the detection model. Given the enormous volume of log data, such as the Spirit dataset containing 272,816,564 entries, memory limitations may arise during processing. To cluster more efficiently, we process the data in chronological

segments. After clustering each segment, the top *n* points from the density center of each cluster are retained and added to the subsequent data segment for re-clustering. This process ensures that only a small subset of the selected data requires labeling. In our experiments, the datasets are pre-labeled, so we simply retrieve the corresponding labels for the selected data.

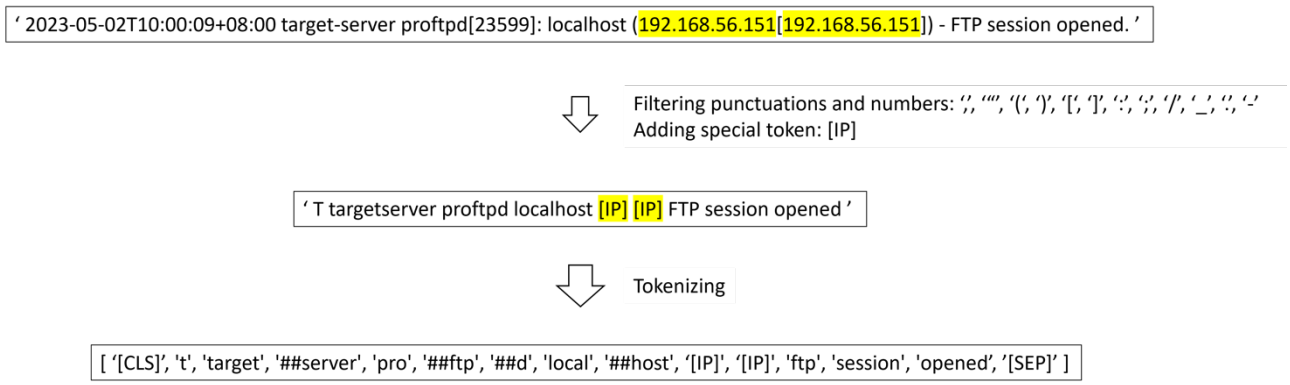


Fig. 3: The example of data preprocessing and tokenization.

In the subsequent anomaly detection phase, the logs are first tokenized and encoded using the BERT tokenizer before computing embeddings with SBERT. A detailed explanation of this method will be provided in the following section.

By utilizing an unsupervised clustering method, this approach avoids the need to predefine the number of clusters and is capable of identifying sequences with similar semantics. This effectively mitigates the issue of handling large volumes of duplicate data and addresses the imbalance between normal and anomalous data. The overall process is illustrated in Fig. 4.

4.4. Anomaly Detection Model

After obtaining and labeling a small subset of training data in the previous stage, this phase uses these data to build the anomaly detection model. To accurately capture contextual information in the logs and effectively handle the variability of log recording formats, we employ the BERT tokenizer, which is based on the WordPiece algorithm [20], to tokenize and encode the preprocessed logs. BERT [21] is then used to compute the semantics and contextual relationships within the logs.

In this step, we first use the BERT tokenizer to tokenize each log sequence and obtain vectorized log data. Subsequently, the pre-trained BERT model is fine-tuned to generate embedding vectors. Finally, a simple classifier is applied to perform anomaly classification based on the semantic and sequential information extracted from the logs.

Due to the strong transferability of the pre-trained BERT model and its ability to learn rich semantic and contextual representations, excellent results can be achieved using only a small amount of high-quality data selected in the previous phase. This approach eliminates the need for a complex classifier architecture, which not only speeds up computation but also reduces the risk of overfitting when training data is limited.

5. Performance Evaluation

To evaluate the performance of the proposed method, we formulate the following research questions:

- RQ1: How effective is the proposed framework?
- RQ2: How robust is the proposed method when handling unknown events?
- RQ3: Do the selected datasets also yield good results when applied to other frameworks?

5.1. Experimental Setup

To evaluate the performance of the proposed method, we utilize five datasets of varying scales and types, including four supercomputer datasets: Blue Gene/L (BG/L), Thunderbird, Spirit, and Liberty [22]. Table 2 provides the number of anomalies and total logs for each dataset. The supercomputer datasets are sourced from Loghub, a platform specializing in providing datasets for log analysis. The datasets we selected are publicly available and have been used in prior research, representing supercomputer systems composed of multiple nodes. Experts in the field labeled anomalies in these datasets.

Model training and testing were performed using PyTorch on a system running Windows 11, equipped with an Intel i7-12700 CPU, 64 GB of RAM, and an NVIDIA RTX 3090 GPU. The window and step sizes for grouping were set to 10 and 1, respectively. The minimum cluster size ranged from 3 to 5, and the data selection threshold was set to 5000. We utilized pre-trained weights for SBERT (all-MiniLM-L6-v2) and BERT (bert-base-uncased). For classification, a single-layer fully connected network with a sigmoid activation function was employed. During training, the BERT model was configured with a dropout rate of 0.4, a learning rate of 3e-5, a weight decay of 0.001, and was trained for 5 epochs with a batch size of 8. The performance of the detection model was evaluated using precision, recall, and F1-score metrics.

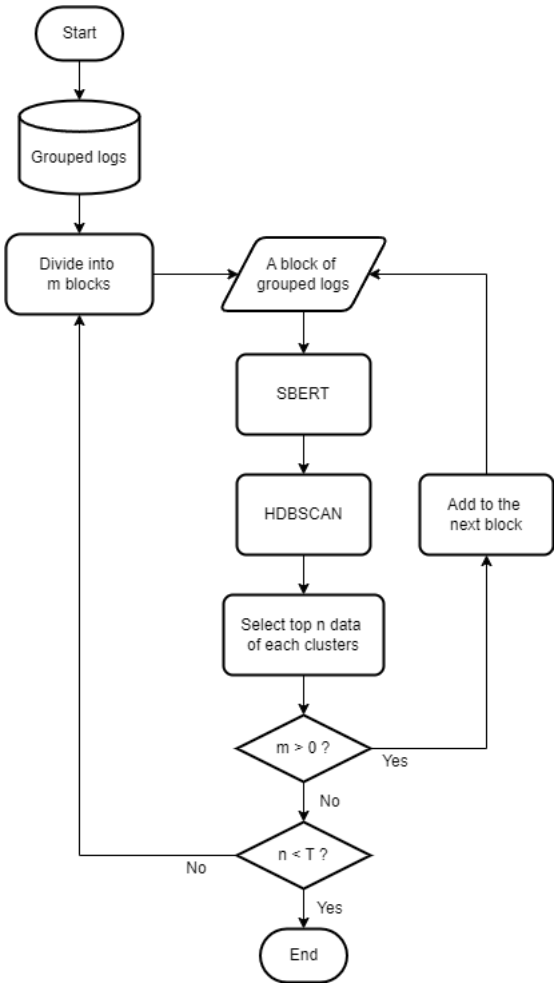


Fig. 4: Flow chart of data selection.

Table 2: The supercomputer datasets

Dataset	# of logs	# of anomaly	% of anomaly
BG/L	4,747,963	348,460	7.34
Spirit	272,298,969	172,816,564	63.47
Thunderbird	211,212,192	3,248,239	1.54
Liberty	265,569,231	97,090,778	36.56

5.2. RQ1: Performance of the Proposed Method

Five datasets were individually inputted into the system for training and testing to evaluate the performance of the proposed detection system. Each dataset was first split chronologically into training and testing sets using an 80:20 ratio. Table 3 shows the amount of data used for training and testing across the five datasets, along with the results after data selection. It is important to note that labels were not required during the data selection process. Fig. 5 shows the testing results after training the detection model on the selected data. The

results demonstrate that our system successfully selected less than 0.01% of the data from millions of entries in the supercomputer datasets, while achieving an F1-score exceeding 0.96.

By clustering semantically similar log windows, redundancy is minimized, optimizing model training. This approach selects semantically diverse events, reducing duplicate data and lowering training costs, all while remaining unaffected by the original binary class distribution.

Table 3: Statistics of datasets and results of data selection

Dataset	Data type	# of training data	# of selected data	# of testing data
BG/L	Anomaly	333,610	148	56,646
	Total	3,798,363	2,960	949,591
Spirit	Anomaly	144,282,168	159	41,540,582
	Total	220,000,000	645	52,298,960
Thunderbird	Anomaly	8,941,359	529	4,513,438
	Total	168,000,000	4,591	43,212,183
Liberty	Anomaly	117,785,504	3,132	25,015,575
	Total	208,000,000	4,319	57,569,222

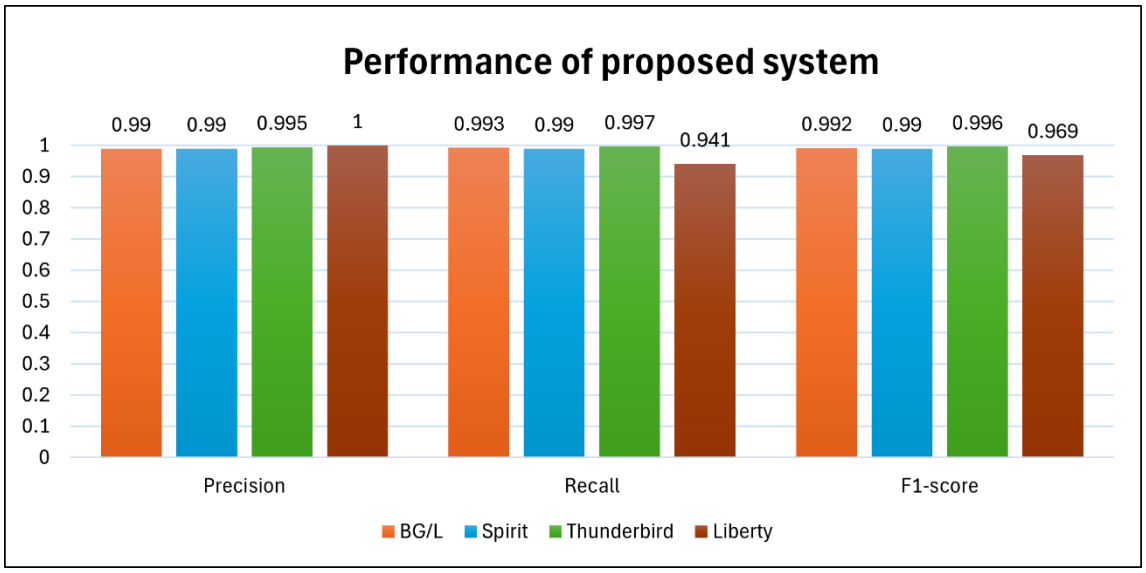


Fig. 5: Performance of the proposed method.

5.3. RQ2: Robustness of the Proposed Method

To evaluate the robustness of the proposed detection system, an in-depth analysis was conducted on the supercomputer dataset by testing its ability to identify unknown events. Specific data were removed from the dataset to create this scenario. First, both anomalous and normal events were selectively removed from the dataset, and the detection system was then trained on the remaining data. Anomalous events were removed from categories with fewer than 5,000 entries in the original dataset. For normal events, 15 event templates were randomly selected using Drain, and logs associated with these templates were removed. The results, shown in Fig. 6, indicate that even when encountering unknown events, the proposed system achieved an F1-score exceeding 0.91.

Since the supercomputer dataset contains anomalous events that often include explicit anomalous semantics, such as "*rts panic! - stopping execution.*" Therefore, our method can determine the normality or anomaly of unknown events by learning the semantic patterns in the existing training data. The studies in [23] and [24] also highlight that, compared to context-based anomaly detection, such datasets often contain more

point anomalies with clear semantic features. In conclusion, our proposed method effectively handles unknown events, demonstrating the robustness of the system.

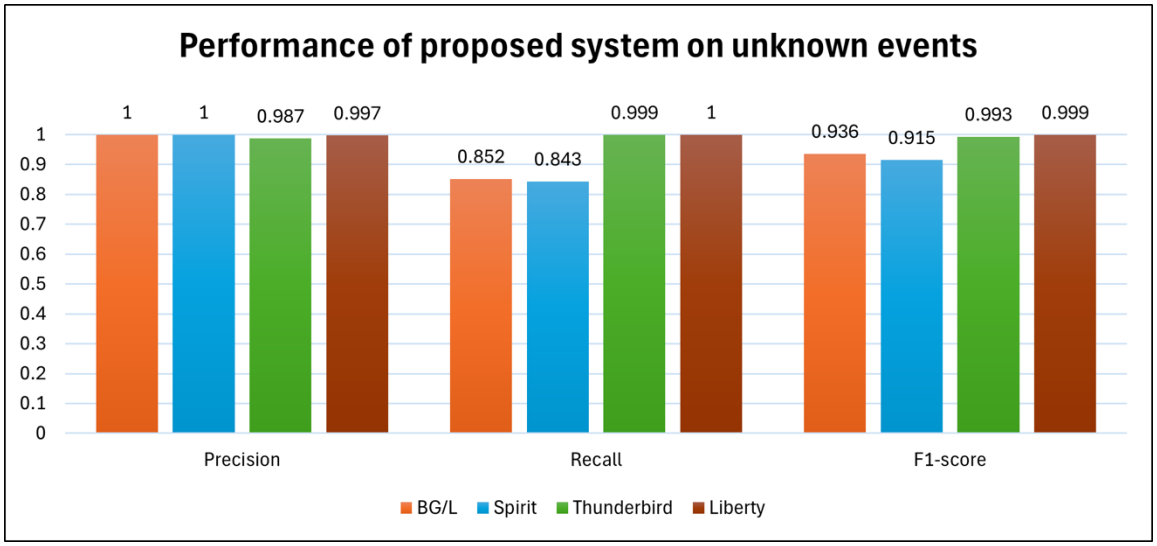


Fig. 6: Performance of the proposed method in detecting unknown events.

5.4. RQ3: Performance of Other Methods Using Selected Data

Finally, to assess the effectiveness of the proposed data selection method in other anomaly detection approaches, we selected three related studies: NeuralLog [11], LogBERT [13], and PLELog [14]. The data selected by our method was used as the initial training set for these models.

NeuralLog is a method that does not rely on log parsing. LogBERT employs a BERT-based pre-training approach, introducing masked log key prediction (MLKP) and utilizing Volume of Hypersphere Minimization (VHM) to control the distribution of normal and anomalous events. PLELog begins by extracting a small subset of normal logs from the original data as initial training data, then uses HDBSCAN to probabilistically estimate labels for the remaining unlabeled data. This probabilistically labeled data is subsequently used for model training. Both LogBERT and PLELog are semi-supervised learning models, so only the normal events from the selected data were used as their initial training set.

Fig. 7 shows the results of the three methods using our selected data. All three methods achieved an F1-score of over 0.8 across each dataset, with NeuralLog showing the best performance. This may be due to NeuralLog's fully supervised training, which facilitates more accurate anomaly detection than semi-supervised approaches. However, NeuralLog is significantly time-consuming, requiring approximately 3 hours to process 5,000 inputs. In contrast, our method can process tens of thousands of inputs in under 10 seconds. LogBERT, which only uses normal data for training, flags a log template as anomalous if it does not appear in the predicted candidate set. This approach may result in lower accuracy when dealing with more ambiguous templates. The performance of PLELog is influenced by the selected log intervals; while inputting the entire dataset can improve accuracy, it also leads to increased training costs.

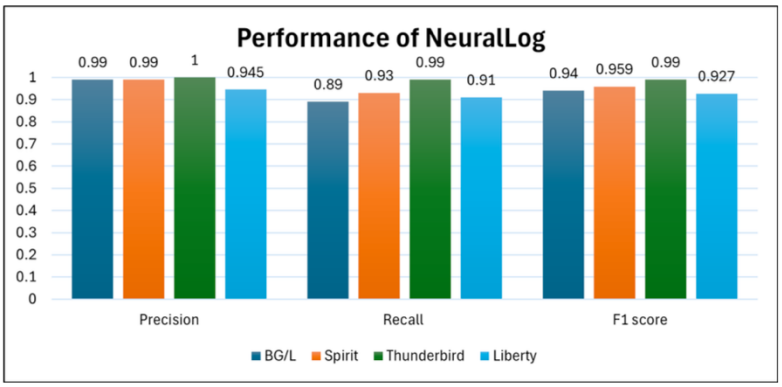
In summary, while NeuralLog offers higher accuracy, it is significantly more time-consuming. LogBERT has limitations in recognizing ambiguous templates, and PLELog's accuracy is highly dependent on data selection, with increased training costs. Both LogBERT and PLELog are also affected by the reliance on log parsing. In contrast, our method demonstrates strong capabilities for processing large volumes of data efficiently, with excellent robustness and practical applicability.

6. Conclusion

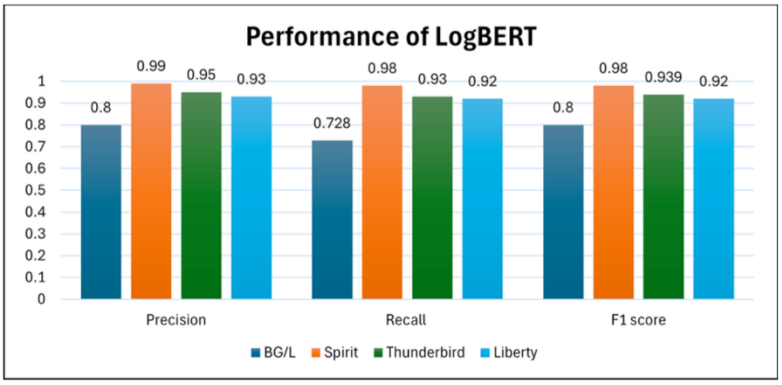
In recent years, log-based anomaly detection systems have gained significant attention. However, building effective detection models presents challenges, such as managing large volumes of highly repetitive log data and dealing with the instability of log recording methods.

This study addresses these challenges by proposing a relevance-based data selection method that reduces data labeling requirements while maintaining high performance. The method leverages regular expressions

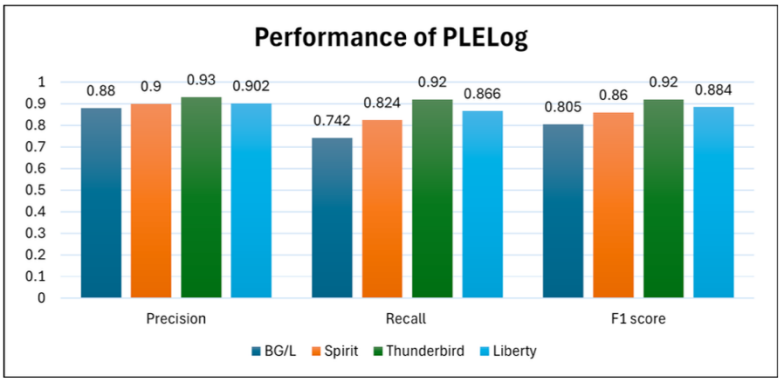
and a WordPiece-based tokenizer to minimize model sensitivity to log variations, thereby enhancing robustness. BERT is employed for semantic vectorization, and a classifier is used to identify normal and anomalous events. Experimental results show an F1-score exceeding 0.96 on supercomputer datasets and over 0.91 for unknown event detection. Furthermore, applying this method as initial training in other studies resulted in F1-scores above 0.8, demonstrating its clear advantage.



(a) Performance of NeuralLog.



(b) Performance of LogBERT.



(c) Performance of PLELog.

Fig. 7: Performance of 3 related works using our selected data.

7. Acknowledgements

This work was partially supported by NSTC of Taiwan under Grant number NSTC 111-2221-E-A49 -193 -MY3.

8. References

[1] V. Chandola, A. Banerjee, and V. Kumar, “Anomaly detection: A survey,” ACM computing surveys (CSUR), vol. 41, no. 3, pp. 1–58, 2009.

[2] C. Yuan, N. Lao, J.-R. Wen, J. Li, Z. Zhang, Y.-M. Wang, and W.-Y. Ma, “Automated known problem diagnosis with event traces,” ACM SIGOPS Operating Systems Review, vol. 40, no. 4, pp. 375–388, 2006.

[3] W. Xu, L. Huang, A. Fox, D. Patterson, and M. I. Jordan, “Detecting large-scale system problems by mining console logs,” in Proceedings of the ACM SIGOPS 22nd symposium on Operating systems principles, 2009, pp. 117–132.

[4] M. Du, F. Li, G. Zheng, and V. 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, 2017, pp. 1285–1298.

[5] W. Meng, Y. Liu, Y. Zhu, S. Zhang, D. Pei, Y. Liu, Y. Chen, R. Zhang, S. Tao, P. Sun et al., “Loganomaly: Unsupervised detection of sequential and quantitative anomalies in unstructured logs.” in IJCAI, vol. 19, no. 7, 2019, pp. 4739–4745.

[6] V.-H. Le and H. Zhang, “Log-based anomaly detection with deep learning: How far are we?” in Proceedings of the 44th international conference on software engineering, 2022, pp. 1356–1367.

[7] P. He, J. Zhu, Z. Zheng, and M. R. Lyu, “Drain: An online log parsing approach with fixed depth tree,” in 2017 IEEE international conference on web services (ICWS), pp. 33–40.

[8] M. Du and F. Li, “Spell: Streaming parsing of system event logs,” in 2016 IEEE 16th International Conference on Data Mining (ICDM), pp. 859–864.

[9] A. Oliner and J. Stearley, “What supercomputers say: A study of five system logs,” in 37th annual IEEE/IFIP international conference on dependable systems and networks (DSN’07), 2007, pp. 575–584.

[10] M. Landauer, S. Onder, F. Skopik, and M. Wurzenberger, “Deep learning for anomaly detection in log data: A survey,” Machine Learning with Applications, vol. 12, p. 100470, 2023.

[11] V.-H. Le and H. Zhang, “Log-based anomaly detection without log parsing,” in 2021 36th IEEE/ACM International Conference on Automated Software Engineering (ASE). IEEE, 2021, pp. 492–504.

[12] X. Zhang, Y. Xu, Q. Lin, B. Qiao, H. Zhang, Y. Dang, C. Xie, X. Yang, Q. Cheng, Z. 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, 2019, pp. 807–817.

[13] H. Guo, S. Yuan, and X. Wu, “Logbert: Log anomaly detection via bert,” in 2021 international joint conference on neural networks (IJCNN), 2021, pp. 1–8.

[14] L. Yang, J. Chen, Z. Wang, W. Wang, J. Jiang, X. Dong, and W. Zhang, “Semi-supervised log-based anomaly detection via probabilistic label estimation,” in 2021 IEEE/ACM 43rd International Conference on Software Engineering (ICSE), 2021, pp. 1448–1460.

[15] S. Liu, L. Deng, H. Xu, and W. Wang, “Logbd: A log anomaly detection method based on pretrained models and domain adaptation,” Applied Sciences, vol. 13, no. 13, p. 7739, 2023.

[16] S. Huang, Y. Liu, C. Fung, H. Wang, H. Yang, and Z. Luan, “Improving log-based anomaly detection by pre-training hierarchical transformers,” IEEE Transactions on Computers, vol. 72, no. 9, pp. 2656–2667, 2023.

[17] S. Lu, X. Wei, Y. Li, and L. Wang, “Detecting anomaly in big data system logs using convolutional neural network,” in 2018 IEEE 16th Intl. Conf. on Dependable, Autonomic and Secure Computing, 16th Intl. Conf. on Pervasive Intelligence and Computing, 4th Intl. Conf. on Big Data Intelligence and Computing and Cyber Science and Technology Congress (DASC/PiCom/DataCom/CyberSciTech), 2018, pp. 151–158.

[18] N. Reimers and I. Gurevych, “Sentence-bert: Sentence embeddings using siamese bert-networks,” arXiv preprint arXiv:1908.10084, 2019.

[19] L. McInnes, J. Healy, S. Astels et al., “hdbscan: Hierarchical density based clustering.” J. Open Source Softw., vol. 2, no. 11, p. 205, 2017.

[20] Y. Wu, M. Schuster, Z. Chen, Q. V. Le, M. Norouzi, W. Macherey, M. Krikun, Y. Cao, Q. Gao, K. Macherey et al., “Google’s neural machine translation system: Bridging the gap between human and machine translation,” arXiv preprint arXiv:1609.08144, 2016.

[21] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, “Bert: Pre-training of deep bidirectional transformers for

language understanding," arXiv preprint arXiv:1810.04805, 2018.

[22] J. Zhu, S. He, P. He, J. Liu, and M. R. Lyu, "Loghub: A large collection of system log datasets for ai-driven log analytics," in 2023 IEEE 34th International Symposium on Software Reliability Engineering (ISSRE), 2023, pp. 355–366.

[23] T. Wittkopp, P. Wiesner, D. Scheinert, and O. Kao, "A taxonomy of anomalies in log data," in International Conference on Service-Oriented Computing. Springer, 2021, pp. 153–164.

[24] J. Qi, Z. Luan, S. Huang, C. Fung, H. Yang, H. Li, D. Zhu, and D. Qian, "Logencoder: Log-based contrastive representation learning for anomaly detection," IEEE Transactions on Network and Service Management, vol. 20, no. 2, pp. 1378–1391, 2023.