Pertanika J. Sci. & Technol. 33 (S4): 161 - 177 (2025)

PERTANIKA JOURNALS

SCIENCE & TECHNOLOGY

Journal homepage: http://www.pertanika.upm.edu.my/

Differences Between Research Log Datasets and Development Field Logs and the Creation of the Complexity Evaluation Index

Hironori Uchida[1]*, Keitaro Tominaga[2], Hideki Itai[2], Yujie Li[1] and Yoshihisa Nakatoh[1]

[1]*Kyushu Institute of Technology, Kitakyushu, 804-8550, Japan*
[2]*Panasonic System Design Co., Ltd, Yokohama, 222-0033, Japan*

ABSTRACT

In the industrial domain, logs are widely applied in the management and maintenance of software systems to ensure reliability and availability. Furthermore, in the research field, various deep learning methods such as CNNs, LSTMs, and Transformers have been reported to achieve high accuracy in anomaly detection studies. However, there are challenges to their adoption in development fields. One reason is the limited datasets used in research, which lack a comprehensive evaluation for general applicability. To address this, we have prepared metrics to assess the complexity of log datasets necessary for creating a log generator for research purposes. We conducted a comparative study on the complexity of datasets in both research and industrial domains. Our evaluation of log sequence complexity, using frequency of occurrence and the Gini coefficient, showed that industrial logs are more complex across all metrics. This highlights the increased need for datasets close to the industrial domain for research purposes. Our study's findings suggest that a clear metric for dataset complexity can be achieved by converting logs into templates and then into sequences of size 10, evaluated using the Gini coefficient or kurtosis. Future work will involve developing a generator that produces logs close to those found in development environments, using these metrics as target values.

Keywords: Anomaly detection, complexity, evaluation index, log generator, system log

ARTICLE INFO

Article history:
Received: 01 April 2024
Accepted: 10 April 2025
Published: 10 June 2025

DOI: https://doi.org/10.47836/pjst.33.S4.09

E-mail address:
uchigzi@outlook.com (Hironori Uchida)
tominaga.keitaro@jp.panasonic.com (Keitaro Tominaga)
itai.hideki@jp.panasonic.com (Hideki Itai)
yzyjli@gmail.com (Yujie Li)
nakatoh@ecs.kyutech.ac.jp (Yoshihisa Nakatoh)
*Corresponding author

INTRODUCTION

Logs record vital information during system execution. In the industrial domain, particularly within large-scale systems, logs are extensively used to manage and maintain software systems to ensure reliability and

e-ISSN: 2231-8526

availability. In the research field, various deep learning methods such as CNNs (Du et al., 2017; Lu et al., 2018), LSTM (Meng, et al, 2019; Zhu et al, 2020), and Transformers (Guo et al, 2022; Nedelkoski et al, 2021) have been applied to anomaly detection studies. While these studies report high accuracy, there are challenges in their adoption in development fields (Le et al, 2022). One commonly used dataset in log anomaly detection research is Loghub (Zhu et al, 2023), which includes logs from multiple operating systems and applications such as HDFS (a distributed system), BGL (a supercomputer), and ThunderBird (a supercomputer). However, Loghub has only a few types of labeled logs, and there are only one or two instances of each type. This has raised concerns about the scarcity of datasets (He et al, 2022). For example, a study investigating the contents of the BGL dataset reported that specific logs were streaming continuously, and a particular log sequence pattern (e.g., logs extracted with a window size of 10 and stride of 1) accounted for about 40% of the total (Uchida et al, 2023). While this trend is frequent in system and server logs, it significantly differs from logs produced by systems comprising applications and operating systems that form a large part of societal systems. This discrepancy suggests a potentially significant difference in the complexity of anomaly detection problems.

Therefore, we believe that the choice of dataset is crucial for advancing log anomaly detection research that is also effective in development fields. However, publicly sharing logs from the development and post-development stages is challenging due to various rights. Our research aims to create logs as close as possible to those found in development fields.

In this study, we conducted the following investigations to develop a log dataset generator that closely resembles those in development environments:

1. Development of evaluation metrics to assess the complexity of log datasets.
2. Investigation of the differences in complexity between research log datasets and those from development environments using the metrics developed in point 1.

METHODS

Datasets

This discussion introduces the research dataset and the logs from the development fields used in this experiment.

Dataset in Research Fields

The dataset in the research fields utilized the Loghub dataset, commonly used in studies on log anomaly detection. Loghub contains a variety of datasets, such as operating systems and servers. We selected "Mac", "Linux", "Windows", "Android_v1", "BGL", and "Thunderbird" for our study. We applied a restriction for datasets exceeding 5,000,000 lines by removing logs beyond the 5,000,000th line.

Logs in the Development Fields

The logs from the development environment were extracted from a system presently undergoing development. This system is composed of multiple applications operating on a specific operating system, and the logs analyzed in this study originated from the output of these diverse systems. Specifically, the three logs utilized were denoted as "Dev_v1," "Dev_v2," and "Dev_v3," respectively.

Evaluation Metrics for Measuring Complexity

This discussion introduces the metrics used in this experiment to evaluate log complexity. As there is no standardized metric for assessing the complexity of logs, we compiled various metrics used across different datasets. Additionally, we set the following conditions for complexity in this experiment: If specific logs are continuously output or the number of specific logs is disproportionately high compared to the total logs, we consider the dataset's complexity low.

This is because complex systems that include an OS and multiple applications, specific logs are not continuously output; instead, various types of logs are generated concurrently. Thus, a balanced distribution of different types of logs indicates higher complexity. Under this condition, we used the frequency of each log type to measure complexity using various metrics.

Evaluation Metric 1: Number of Logs Per Second

One of the metrics we used is the number of logs per second. We assumed that the more logs and types per second, the more complex the system and the dataset. To determine whether a dataset is complex because of a high number of logs at specific times and due to a consistently high number of logs, we investigated the number of logs per second.

We calculated using the time part of each log format. Furthermore, as complexity metrics, we used the calculated number of logs per second to find the average, population standard deviation (pstdev), median, maximum, and minimum values.

The evaluation of complexity for each metric is as follows:

1. Mean and Standard Deviation: If the average is high and the standard deviation is low, it indicates that the variance in the number of logs per second is small and the average is high, suggesting a high complexity without bias.
2. Max, Min, Mean, Median: If there is a significant difference between the max and min values, and they are far from the mean and median, it indicates high complexity.

Evaluation Metric 2: Percentage of Data Types in Total Data Count (PDT)

Log type refers to the number of different types of data. In this study, there are three main data types: (1) original logs, (2) templates, and (3) sequence data. The "percentage of data types in total data count" is evaluated by dividing the total number of types by the total number of data points for these data types. A higher value indicates the presence of a larger variety of data types, suggesting higher complexity in the system.

Evaluation Metric 3: Frequency of Occurrence

We calculate the frequency of occurrence for each log type and use the average and standard deviation of these frequency values as complexity indicators. If the variation in frequency values is small, it is assumed that a variety of logs are being produced in large quantities, indicating high complexity.

Evaluation Metric 4: Kurtosis

Kurtosis is a statistical measure that characterizes the shape of a probability distribution, measuring the thickness of the tails and the sharpness of the central peak of the distribution. It allows us to assess how sharp the peak of the distribution is and how thick the tails are (DeCarlo et al., 1997).

In this experiment, kurtosis is calculated using the frequency of occurrence of each log as input. As a preprocessing step, we prepare two sequences of frequency values: one sorted in ascending order and the other in descending order, and then concatenate them to form a convex graph. This process enables the measurement of the kurtosis of frequency occurrence. A kurtosis greater than zero indicates that the distribution is sharper than a normal distribution, suggesting that certain logs occur frequently and there is bias in the data. Conversely, if kurtosis is less than zero, the distribution is not as sharp as a normal distribution and has wider tails, indicating less bias in frequency occurrence. The calculation formula (Fisher's definition) is as Equation 1:

$$Kurtosis = \frac{N(N-1)}{(N-1)(N-2)(N-3)} \sum_{i}^{N} \left(\frac{x_i - \bar{x}}{x}\right)\left(\frac{x_i - \bar{x}}{s}\right)^4 - \frac{3(N-1)^2}{(N-1)(N-3)} \quad [1]$$

where N represents the sample size, x_i represents each data point, $\bar{x}$ is the sample mean, and s is the sample standard deviation. In this definition, the kurtosis of a normal distribution is 0.

Evaluation Metric 5: Gini Coefficient

The Gini coefficient is a measure of inequality commonly used in economics, ranging from 0 (complete equality) to 1 (complete inequality) (Sen et al., 1973). In this experiment, the Gini coefficient is calculated using the frequency of occurrence of each log as input. When the Gini coefficient is close to 0 (complete equality), it indicates a low bias in frequency of occurrence, suggesting high complexity in the dataset. Conversely, when the Gini coefficient is close to 1 (complete inequality), it indicates a high bias in the dataset, suggesting low complexity. In this experiment, we used the simplified formula in Equation 2:

$$GiniCoefficient = \frac{2\sum_{i=1}^{n} i x_i}{n\sum_{i=1}^{n} x_i} - \frac{n+1}{n} \quad [2]$$

where x_i represents the sorted data, i represents the rank of the data (in sorted order), and n represents the total number of data points.

Evaluation Metric 6: Entropy

In the field of information theory, entropy is used to represent the uncertainty of data (Shannon et al., 1948). In this experiment, entropy is calculated using the frequency of occurrence of each log as input. High entropy indicates that the frequency of occurrence is relatively evenly distributed, suggesting high complexity. Low entropy indicates that a few logs occur frequently, suggesting low complexity. As a preprocessing step, the frequency of occurrence of each log is converted into a probability by dividing by the total frequency of occurrence. The preprocessing and the formula for calculating entropy are as Equation 3:

$$Entropy = -\sum p(x_i) * log_2\big(p(x_i)\big) \quad [3]$$

where $p(x_i)$ represents the probability of each data point.

Evaluation Metric 7: Mean Absolute Deviation (MAD)

This metric represents how far data points are from the mean value. MAD, like the mean, indicates the central tendency of data but is less influenced by outliers (Huber et al. 1981). A small MAD indicates that the frequency of occurrence of logs is average, suggesting high complexity in the dataset (Equation 4):

$$Mean\ Absolute\ deviation = \frac{1}{N}\sum_{i=1}^{N} |x_i - \bar{x}| \quad [4]$$

where N is the size of the dataset, x_i represents each data point, and $\bar{x}$ is the mean value of the dataset.

Experimental Procedure

To explore appropriate methods for investigating the complexity of log datasets, this study processed logs and converted them into the following five types of data analysis:

1. Original logs
2. Histogram of original logs (frequency of occurrence)
3. Histogram of log sequence data (frequency of occurrence)
4. Histogram of log templates (frequency of occurrence)
5. Histogram of log template sequence data (frequency of occurrence)

Experimental Procedure 1: Original Logs

In the complexity evaluation experiment of the original logs, we extract the contents part of the logs produced by each system and treat them as different logs if even one character in the string varies. For example, as shown in Table 1, logs Id1 to Id3 are considered different logs even though they only differ in parameters (numbers). This experiment investigates Metrics 1 (number of logs per second) and Metric 2.

Table 1
Examples of original logs

Id	Log
1	ddr: activating redundant bit steering: rank=0 symbol=25
2	ddr: activating redundant bit steering: rank=0 symbol=9
3	ddr: activating redundant bit steering: rank=0 symbol=23
4	1 ddr errors(s) detected and corrected on rank 0, symbol 2, bit 5
5	1 ddr errors(s) detected and corrected on rank 0, symbol 2, bit 0
6	30 ddr errors(s) detected and corrected on rank 0, symbol 9, bit 6

Experimental Procedure 2: Histogram of Original Logs (Frequency of Occurrence)

In this experiment, the complexity of original logs is evaluated by extracting the contents part of logs from each system and treating them as different logs if even one character in the string varies. Metrics 3 to 7 are investigated using the frequency of occurrence for logs.

Experimental Procedure 3: Log Sequence Data

The data uses the contents part of logs produced by each system, similar to Experimental Procedure 1. This data is converted into sequence data using a windowing process

(Window=10, Slide=1) from the chronological order in which the logs are output. These size 10 log sequence data are treated as one piece of data, and those that differ as sequences, as shown in Table 2, are treated as different data. Metrics 3 to 7 are investigated.

Table 2
Examples of log sequence data

Id	Log Sequence Data
1	[2, 2, 2, 2, 2, 2, 2, 2, 2, 2]
2	[2, 2, 2, 2, 2, 2, 2, 2, 2, 4]
3	[3, 3, 2, 2, 3, 2, 3, 2, 3, 2]

Experimental Procedure 4: Log Templates

Log templates extract and represent the common structure of the contents part of logs in a reusable format. They are commonly used as inputs for log analysis tools and deep learning models for anomaly detection. Templates differentiate between the variable parts (variables) and constant parts (constants) in a message, as shown in Table 3, with variable parts represented by <*>. In this study, we used Drain (He et al., 2017), a highly accurate tool for extracting parameters and creating templates. Drain employs a tree-structured learning technique with several parameters, including a threshold. For this experiment, we used the same parameters as those in a paper that investigated the accuracy of various DL models for log anomaly detection (Chen et al., 2022). The parameters used are shown in Table 4. Experimental Procedure 4 investigates Metrics 2 to 7.

Table 3
Examples of log templates

Id	Log Template
1	ddr: activating redundant bit steering: rank=0 <*>
2	<*> ddr errors(s) detected and corrected on rank 0, symbol <*> bit <*>
3	CE sym <*> at <*> mask <*>

Table 4
Drain parameters used in this study

Dataset type	Regex	Similarity threshold	Depth of all leaf nodes
BGL	[r'core\.\d+']	0.5	4
Android	[r'(/[\w-]+)+', r'([\w-]+\.){2,}[\w-]+', r'\b(\-?\+?\d+)\b|\b0[Xx][a-fA-F\d]+\b|\b[a-fA-F\d]{4,}\b']	0.2	6
Thunderbird	[r'(\d+\.){3}\d+']	0.5	4
Windows	[r'0x.*?\s']	0.7	5
Linux	[r'(\d+\.){3}\d+', r'\d{2}:\d{2}:\d{2}']	0.39	6
Mac	[r'([\w-]+\.){2,}[\w-]+']	0.7	6

Experimental Procedure 5: Sequence Data of Log Templates

Similar to Experimental Procedure 3, the data uses the contents of the logs produced by each system, and converts them into templates. These templates are converted into sequence data using a windowing process (Window=10, Slide=1) from the chronological order in which the templates are output. These size 10 template sequence data are treated as one piece of data, and those that differ as sequences, as shown in Table 5, are treated as different data. Metrics 3 to 7 are investigated.

Table 5
Examples of log template sequence data

Id	Log Template Sequence Data
1	[47, 47, 47, 47, 47, 47, 47, 47, 47, 26]
2	[47, 47, 47, 47, 47, 47, 47, 47, 47, 47]
3	[47, 47, 47, 47, 47, 48, 48, 48, 48, 48]

RESULTS

Experiment 1: Original Logs

We assessed the information quantity introduced in Experimental Procedure 1 using evaluation criteria 1 to 2. According to the definition original logs are treated as distinct logs if they do not fully match; each log is assigned a unique ID.

The results for each system are shown in Table 6. The criteria for high complexity include a high mean value, a small difference between the mean and median values, and small differences between the mean and the maximum and minimum values. The datasets that most closely meet these criteria are BGL and the three development field datasets. Linux has a small variance in the number of logs per second, but it is considered less complex

Table 6
Results of evaluation metrics 1 and 2 in Experiment 1

Dataset Name	Number of logs per second				PDT	Component
	mean	pstdev	median	max/min		
Mac	5.633	31.524	2	399 / 7	***0.394***	126
Linux	***1.658***	***2.844***	***1***	***53 / 1***	***0.442***	72
Windows	906.668	3855.942	18	35411 / 1	0.042	13
Android_v1	75.538	166.253	26	2568 / 1	0.180	1756
BGL	15.744	22.204	7	393 / 1	0.076	14
Thunderbird	7.862	58.545	3	10246 / 1	0.083	173
Dev_v1	402.952	402.509	163	1663 / 2	0.352	765
Dev_v2	***61.113***	***78.237***	***36***	***379 / 1***	0.375	11
Dev_v3	21.520	48.225	2	304 / 1	0.793	106

due to the low average number of logs. Other datasets have a large standard deviation and a significant difference between the mean and maximum values, indicating that logs are produced in large quantities at specific times, leading to a lower complexity rating.

Next, we summarize the results for the metric ratio of types to total log count. This metric investigates the diversity of log types within the dataset. Among the systems studied, Mac, Linux, and the three development field systems show relatively less bias in the types of logs.

Experiment 2: Histogram of Original Logs (Frequency of Occurrence)

We assessed the information quantity introduced in Experimental Procedure 1 based on evaluation criteria 3 through 7. According to the definition that original log histograms treat each log as a distinct entity unless they are an exact match, the histograms of each log are considered the sources of information.

Tables 7 and 8 show the results for each system. A notable result across all systems is that the median value is close to 1, indicating that most logs are infrequently produced (Table 7). Systems with a small difference between the mean and maximum values and a small standard deviation are Mac, Linux, and the three development field systems.

The systems with high complexity, as listed in Table 8, are summarized below:

1. Kurtosis: Mac, Linux, and the three development field systems are relatively complex. Notably, Development v2 and v3 show even less bias in frequency of occurrence compared to the other three.
2. Gini coefficient: Development v3 shows significantly less bias compared to the others.
3. Entropy: The results were almost identical across all systems.
4. Mean absolute deviation: Systems with relatively less bias are Mac, Linux, and the three development field systems.

Table 7
Results of evaluation metrics 3 in Experiment 2

Dataset Name	Histogram				
	mean	pstdev	median	max	min
Mac	2.537795	28.07699	1	2397	1
Linux	2.262353	11.82191	1	1043	1
Windows	23.86387	140.1128	4	34668	1
Android_v1	5.566029	131.1742	1	27822	1
BGL	13.1531	498.9645	1	152734	1
Thunderbird	12.08394	1126.886	1	382340	1
Dev_v1	2.838861	24.58524	1	2165	1
Dev_v2	2.663655	21.73202	1	387	1
Dev_v3	1.261637	1.853959	1	42	1

Table 8
Results of evaluation metrics 4 to 7 in Experiment 2

Dataset Name	Kurtosis	Gini coeff	Entropy	Mean abs dev
Mac	***2138.922004***	***0.597588***	11.96984	***2.798122***
Linux	5400.675048	***0.521918***	11.93911	***2.095992***
Windows	22940.91372	0.84988	***14.62374***	34.60172
Android_v1	14445.56889	0.811196	***12.55274***	8.451361
BGL	50994.03783	0.871437	***13.26365***	19.65518
Thunderbird	57462.50936	0.916109	8.629199	21.82688
Dev_v1	4434.076893	0.625783	11.23046	3.178633
Dev_v2	***243.0334853***	0.619438	7.827793	3.143811
Dev_v3	***359.0682511***	***0.198355***	9.90051	0.474979

Experiment 3: Log Sequence Data

We assessed the information quantity introduced in Experimental Procedure 3 using evaluation criteria 3 to 7. The log sequence data refers to sequence data generated by applying window processing to the logs in chronological order.

Looking at Table 9, it is observed that the three Development Field datasets have a small difference between the mean and standard deviation, indicating a low bias in frequency of occurrence. The systems with high complexity listed in Table 10 are summarized below:

1. Kurtosis: Linux, Windows, and the three Development Field systems show low values, indicating relatively high complexity.
2. Gini Coefficient: Mac and Development v1, in particular, show significantly less bias compared to others. Compared with the results of the log data in Experiment 2, the values for Mac, Linux, Development v1, and v2 have significantly decreased. The evaluation of log sequences measures whether the same logs are being output consecutively, so systems with low values indicate that logs are output in various orders, suggesting high complexity for anomaly detection datasets.
3. Entropy: The results were almost identical across all systems.
4. Mean Absolute Deviation: Systems with relatively less bias include Mac, Linux, and the three Development Field systems.

Table 9
Results of evaluation metrics 3 in Experiment 3

Dataset Name	Histogram of Log Sequence Data				
	mean	pstdev	median	max	min
Mac	25.692	186.8841	1	7685	1
Linux	50.912	501.259	8	10404	1

Table 9 *(continue)*

Dataset Name	Histogram of Log Sequence Data				
	mean	pstdev	median	max	min
Windows	1412.275	42682.57	1	1750733	1
Android_v1	99.82051	824.7748	3	40942	1
BGL	2550.591	42936.94	1	1706751	1
Thunderbird	4928.063	32883.25	45	660553	1
Dev_v1	8.265135	50.03262	1	2165	1
Dev_v2	88.84932	356.4681	2	2081	1
Dev_v3	2.655405	9.642574	1	147	1

Table 10
Results of evaluation metrics 4 to 7 in Experiment 3

Dataset Name	Kurtosis	Gini coeff	Entropy	Mean abs dev
Mac	29835.69004	***0.023354***	***16.56222***	***0.047615***
Linux	696.3898193	0.159301	14.02642	0.371192
Windows	***266.4154937***	0.813842	15.48659	18.17317
Android_v1	188736.1726	0.199083	***19.14314***	0.48477
BGL	397543.7988	0.578114	17.0855	2.614236
Thunderbird	3040665.057	0.297249	***20.00339***	0.817167
Dev_v1	1893.53867	***0.089784***	14.95477	***0.194578***
Dev_v2	***244.9009545***	0.946071	12.66111	46.71347
Dev_v3	***472.3838329***	***0.141798***	10.59842	0.321129

Experiment 4: Log Templates

We assessed the information quantity introduced in Experimental Procedure 4 using evaluation criteria 2 to 7. The log template refers to data in which the common components of each log are templated using a Log Parser.

Tables 11 and 12 present the results for each system. The three development field datasets have a small difference between the mean and standard deviation, indicating a low bias in frequency of occurrence (Table 11). Compared to the original logs (Experiment 2), the differences between systems in mean values and standard deviations become more discernible.

The systems with high complexity, as listed in Table 12, are summarized below:

1. Kurtosis: Thunderbird and Development v1, v2 show low values, indicating relatively high complexity.
2. Gini coefficient: Development v1 and Development v2 have significantly less bias compared to others.

3. Entropy: It is observed that the values for Android, Development v1, and Thunderbird are relatively high.
4. Mean absolute deviation: Development v1 and Development v2 show significantly less bias compared to others.

Table 11
Results of evaluation metrics 2 and 3 in Experiment 4

Dataset Name	Histogram of Templates					Template types
	mean	pstdev	median	max	min	
Mac	25.692	186.8841	1	7685	1	4000
Linux	50.912	501.259	8	10404	1	500
Windows	1412.275	42682.57	1	1750733	1	3538
Android_v1	99.82051	824.7748	3	40942	1	14898
BGL	2550.591	42936.94	1	1706751	1	1848
Thunderbird	4928.063	32883.25	45	660553	1	1013
Dev_v1	**8.265135**	**50.03262**	**1**	**2165**	**1**	5137
Dev_v2	**88.84932**	**356.4681**	**2**	**2081**	**1**	73
Dev_v3	2.655405	9.642574	1	147	1	592

Table 12
Results of evaluation metrics 4 to 7 in Experiment 4

Dataset Name	Kurtosis	Gini coeff	Entropy	Mean abs dev
Mac	779.8136527	0.929228	7.674175	***44.10432***
Linux	364.3586554	***0.892429***	4.195259	81.88374
Windows	1585.393851	0.997397	2.692711	2781.577
Android_v1	979.8777736	0.930228	***9.664662***	160.5336
BGL	1348.354829	0.987413	4.268589	4760.467
Thunderbird	***192.1112006***	0.931104	***8.629199***	7907.527
Dev_v1	785.1040542	***0.817377***	***9.223006***	***11.61492***
Dev_v2	***18.75473595***	0.930945	2.475725	157.9384
Dev_v3	***125.3654616***	***0.585244***	7.409572	2.678757

Experiment 5: Sequence Data of Log Templates

We assessed the information quantity introduced in Experimental Procedure 5 using evaluation criteria 3 to 7. The sequence data of log templates refers to data generated by applying window-based grouping to log template data extracted using a Log Parser in chronological order.

Tables 13 and 14 show the results for each system. The three development field datasets have a small difference between the mean and standard deviation, indicating a

low bias in frequency of occurrence (Table 13). Compared to the histogram of templates (Experiment 4), the differences between systems in terms of mean values and standard deviations become even more apparent.

The systems with high complexity, as listed in Table 14, are summarized below:

1. Kurtosis: The three development field systems show low values, indicating relatively high complexity.
2. Gini coefficient: Mac, Development v1, and Development v2 show significantly less bias compared to others.
3. Entropy: Linux, Windows, Thunderbird, and Development Field sub1 show high complexity.
4. Mean absolute deviation: Development v1 and Development v2 show significantly less bias compared to others.

Table 13
Results of evaluation metrics 3 in Experiment 5

Dataset Name	**Histogram**				
	mean	**pstdev**	**median**	**max**	**min**
Mac	1.599437	16.32902	1	3921	1
Linux	4.768928	135.1905	1	9584	1
Windows	231.894	16117.76	1	1663366	1
Android_v1	1.70683	38.48374	1	27045	1
BGL	29.85183	4498.018	1	1679174	1
Thunderbird	4.095916	202.1167	1	128936	1
Dev_v1	1.315677	3.431559	1	202	1
Dev_v2	7.9375	27.24916	1	248	1
Dev_v3	1.029644	0.229087	1	6	1

Table 14
Results of evaluation metrics 4 to 7 in Experiment 5

Dataset Name	**Kurtosis**	**Gini coeff**	**Entropy**	**Mean abs dev**
Mac	51686.09081	***0.363217***	14.66522	***1.076106***
Linux	4728.412527	0.768955	6.918843	6.516396
Windows	10354.70923	0.99097	4.084183	439.5919
Android_v1	292478.0814	0.405236	***17.67513***	1.277801
BGL	123606.2519	0.959574	6.279894	54.35098
Thunderbird	326215.3618	0.73049	***16.12383***	5.316484
Dev_v1	846.0012814	0.234677	14.29795	0.58936
Dev_v2	***30.80765436***	0.813872	7.034099	11.55499
Dev_v3	***176.9035518***	***0.028296***	10.54513	0.058

DISCUSSIONS

Evaluation Metrics

Firstly, regarding the transformation process of the datasets used, it was found that templates show a greater difference in metrics than original logs, making it easier to evaluate complexity. Moreover, converting to histogram sequence data rather than dealing with original logs or templates line by line clarified the differences in complexity. Converting to templates, which treat logs of the same format differing only in parameters, equally suggests that it is crucial to investigate datasets while preserving their significant parts. The reason sequence data showed clearer differences in complexity is likely due to the issue reported in related studies that "research datasets contain many consecutive logs," allowing us to identify biases in the frequency of specific sequence data occurrences.

These results suggest that the sequence data of templates is informative for evaluating dataset complexity, which is necessary for researching machine learning systems usable in the industrial domain.

As for which metrics are suitable, while consideration is needed for the number of data and types of logs regarding the average and variance of occurrence frequencies, it is clear that they can be easily evaluated.

Next, we discuss the development environment logs under the assumption that they are highly complex due to the system's inherent complexity. As shown in the results in Table 14, kurtosis and the Gini coefficient indicate that the development environment logs are more complex. In contrast, entropy produced different results. Based on these findings, kurtosis and the Gini coefficient are deemed appropriate metrics for evaluating complexity.

Kurtosis tends to exhibit extremely high values, particularly when a small number of data categories dominate, which can amplify differences between datasets. However, it is important to recognize that kurtosis primarily indicates how closely a distribution aligns with a normal distribution. Therefore, it should be applied cautiously when evaluating whether individual data logs are evenly distributed on average. While kurtosis is useful for identifying outliers and understanding distributional characteristics, its interpretation requires careful consideration, especially in contexts involving non-normal data distributions (Kim & White, 2004).

In contrast, the Gini coefficient directly measures inequality within a dataset and effectively quantifies the uniformity of data outputs. Gastwirth (1972) noted that the Gini coefficient provides a clear and straightforward metric for determining whether outputs are evenly distributed across categories. This characteristic makes the Gini coefficient particularly well-suited for evaluating the fairness or uniformity of data outputs, thereby complementing kurtosis-based analysis.

By integrating both metrics into the methodology, kurtosis highlights the sharpness of the distribution, whereas the Gini coefficient evaluates its equality. Together, these metrics enable a more comprehensive assessment of data characteristics.

Differences in Data Characteristics: Research Datasets vs. Industrial Logs

The results from all metrics used in this study showed that logs from development fields exhibited higher complexity. This confirms that there is indeed a difference between research datasets and industrial logs. This discrepancy poses a significant challenge for current research in log anomaly detection, highlighting the need to create research datasets closer to those in the industrial domain.

CONCLUSION

In this study, we investigated metrics to evaluate the complexity of datasets. We examined the differences between research datasets and logs from development environments, aiming to create anomaly detection datasets in logs closer to those in industrial domains. Based on our investigations, we found the following approaches to be suitable:

1. Using sequence data derived from the frequency of occurrences in log templates.
2. Using mean, variance, kurtosis, and Gini coefficient based on the frequency of occurrences as evaluation metrics.

Logs from industrial domains showed higher complexity in all evaluation metrics prepared for this study. Similar to findings in reference studies, these results indicate that current research datasets are significantly different from those in industrial domains, underscoring the need for research datasets closer to the industrial context.

Future work will involve creating an automatic log generator that can be used for research, with the evaluation metrics identified in this study serving as target values. Furthermore, in this study, we only evaluated each complexity metric independently. For example, template types and the evaluation of the Gini coefficient for frequency of occurrences in template sequences reflect different aspects of complexity, namely the diversity of template types and the complexity of the log sequences, respectively. Therefore, to measure the overall complexity of a dataset, it is necessary to observe the balance of each complexity, making creating a comprehensive evaluation method a task for future research.

ACKNOWLEDGEMENTS

This work is supported by a grant from Panasonic System Design and JST, Kyutech Research Fellowship, with Grant Number JPMJFS2133. This work was supported by JST, Kyutech Research Fellowship, Grant Number JPMJFS2133.

REFERENCES

Chen, Z., Liu, J., Gu, W., Su, Y., & Lyu, M. R. (2021). *Experience report: Deep learning-based system log analysis for anomaly detection*. arXiv:2107.05908. https://doi.org/10.48550/arXiv.2107.05908

DeCarlo, L. T. (1997). On the meaning and use of kurtosis. *Psychological Methods*, *2*(3), 292. https://doi.org/10.1037/1082-989X.2.3.292

Du, M., Li, F., Zheng, G., & Srikumar, V. (2017). Deeplog: Anomaly detection and diagnosis from system logs through deep learning. In *Proceedings of the 2017 ACM SIGSAC conference on computer and communications security* (pp. 1285-1298). Association for Computing Machinery. https://doi.org/10.1145/3133956.3134015

Gastwirth, J. L. (1972). The estimation of the Lorenz curve and Gini index. *The Review of Economics and Statistics, 54*(3), 306–316. https://doi.org/10.2307/1937992

Guo, H., Lin, X., Yang, J., Zhuang, Y., Bai, J., Zheng, T., Zhang, B., & Li, Z. (2021). *Translog: A unified transformer-based framework for log anomaly detection*. arXiv:2201.00016. https://doi.org/10.48550/arXiv.2201.00016

He, P., Zhu, J., Zheng, Z., & Lyu, M. R. (2017). Drain: An online log parsing approach with fixed depth tree. In *2017 IEEE international conference on web services (ICWS)* (pp. 33-40). IEEE. https://doi.org/10.1109/ICWS.2017.13

He, S., Zhang, X., He, P., Xu, Y., Li, L., Kang, Y., Ma, M., Wie, Y., Dang, Y., Rajmohan, S., & Lin, Q. (2022). An empirical study of log analysis at Microsoft. In *Proceedings of the 30th ACM Joint European Software Engineering Conference and Symposium on the Foundations of Software Engineering* (pp. 1465-1476). Association for Computing Machinery. https://doi.org/10.1145/3540250.3558963

Huber, P. J. (1981). *Robust Statistics*. Wiley.

Kim, T. H., & White, H. (2004). On more robust estimation of skewness and kurtosis. *Finance Research Letters*, *1*(1), 56-73. https://doi.org/10.1016/S1544-6123(03)00003-5

Le, V. H., & Zhang, H. (2022). Log-based anomaly detection with deep learning: How far are we?. In *Proceedings of the 44th International Conference on Software Engineering* (pp. 1356-1367). Association for Computing Machinery. https://doi.org/10.1145/3510003.3510155

Lu, S., Wei, X., Li, Y., & Wang, L. (2018). 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)* (pp. 151-158). IEEE. https://doi.org/10.1109/DASC/PiCom/DataCom/CyberSciTec.2018.00037

Meng, W., Liu, Y., Zhu, Y., Zhang, S., Pei, D., Liu, Y., Chen, Y., Zhang, R., Tao, S., Sun, P., & Zhou, R. (2019), Loganomaly: Unsupervised detection of sequential and quantitative anomalies in unstructured logs. *International Joint Conference on Artificial Intelligence, 19*(7), 4739-4745. https://doi.org/10.24963/ijcai.2019/658

Nedelkoski, S., Bogatinovski, J., Acker, A., Cardoso, J., & Kao, O. (2020). Self-attentive classification-based anomaly detection in unstructured logs. In *2020 IEEE International Conference on Data Mining (ICDM)* (pp. 1196-1201). IEEE. https://doi.org/10.1109/ICDM50108.2020.00148

Sen, A. (1973). *On economic inequality*. Oxford University Press.

Shannon, C. E. (1948). A mathematical theory of communication. *The Bell System Technical Journal*, *27*(3), 379-423. https://doi.org/10.1002/j.1538-7305.1948.tb01338.x

Uchida, H., Tominaga, K., Itai, H., Li, Y., & Nakatoh, Y. (2023). Verification of generalizability in software log anomaly detection models. In V. K. Parimala (Ed.) *Anomaly Detection-Recent Advances, AI and ML Perspectives and Applications* (pp. 7). IntechOpen. https://doi.org/10.5772/intechopen.111938

Zhu, B., Li, J., Gu, R., & Wang, L. (2020). 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* (pp. 1-7). Association for Computing Machinery. https://doi.org/10.1145/3446132.3446415

Zhu, J., He, S., He, P., Liu, J., & Lyu, M. R. (2023). Loghub: A large collection of system log datasets for ai-driven log analytics. In *2023 IEEE 34th International Symposium on Software Reliability Engineering (ISSRE)* (pp. 355-366). IEEE. https://doi.org/ 10.1109/ISSRE59848.2023.00071