IEEE Access
Multidisciplinary : Rapid Review : Open Access Journal

Received 25 February 2024, accepted 20 March 2024, date of publication 10 April 2024, date of current version 10 June 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3387287

SURVEY

Landscape and Taxonomy of Online Parser-Supported Log Anomaly Detection Methods

SCOTT LUPTON[1,2], HIRONORI WASHIZAKI[1], (Member, IEEE), NOBUKAZU YOSHIOKA[3,4], (Member, IEEE), AND YOSHIAKI FUKAZAWA[5], (Member, IEEE)

[1]Department of Computer Science and Communications Engineering, Waseda University, Tokyo 169-8050, Japan
[2]Nomura Securities Company Ltd., Tokyo 100-8130, Japan
[3]Research Institute for Science and Engineering, Waseda University, Tokyo 169-8555, Japan
[4]QAML Inc., Tokyo 102-0074, Japan
[5]Department of Environmental Science, University of Human Environments, Matsuyama 790-0825, Japan

Corresponding author: Scott Lupton (scott.lupton@toki.waseda.jp)

This work was supported in part by the Japan Society for the Promotion of Science (JSPS) KAKENHI Program under Grant 21KK0179 and Grant 23K18470, and in part by the Japan Science and Technology Agency (JST)-Mirai Program under Grant JPMJMI20B8.

ABSTRACT As production system estates become larger and more complex, ensuring stability through traditional monitoring approaches becomes more challenging. Rule-based monitoring is common in industrial settings, but it has limitations. These include the difficulty of crafting rules capable of detecting unforeseen issues and the burden of manually maintaining rule sets. A potential solution to effectively manage complex system states is log anomaly detection. Workflows for log anomaly detection utilize several fundamental components. These include preprocessors for data cleansing, parsers to extract structured information from raw log data, encoding algorithms to convert extracted data into usable model input features, anomaly detection methods to isolate anomalous signals, and feedback mechanisms to incrementally improve model performance. This study explores the current state of research into online parser-supported log anomaly detection methods, investigates recent research trends, compares the performances of parser and anomaly detection methods using common public datasets and metrics, and assesses their performance evolution over time. Additionally, it classifies available methods using a newly introduced taxonomy, highlights current research gaps, and recommends future research directions.

INDEX TERMS Log parsing, log template extraction, online algorithms, anomaly detection.

I. INTRODUCTION

Enterprise production service teams are responsible for ensuring the health of production estates through proactive monitoring and repair. The real-time monitoring of large, integrated system environments, however, is a non-trivial affair. Traditional rule-based approaches to monitoring have many weaknesses. Creating rules capable of detecting unforeseen issues is challenging and the effort required to manually maintain rule sets is significant.

Modern anomaly detection approaches have shown the potential for practical use against many different forms of log targets (e.g., failures [1], security/network intrusions [2], [3], [4], performance degradation indicators [5], [6], etc.). Anomaly detection methods don't require rule creation or maintenance. Also by nature, they are designed to detect events that are out of the ordinary, making them capable of discovering unforeseen issues. For these reasons, they have the potential to improve upon the weaknesses of rule-based approaches.

Log anomaly detection methods come in many forms. Event-based methods attempt to detect log events not seen previously during periods of system normality. Sequence-based methods make predictions of events based on a window of previous ones, and flag those that fall outside their predictions as anomalies. Online log anomaly detection differs from other forms of anomaly detection in that the input data used is highly unstructured, oftentimes inconsistent in

The associate editor coordinating the review of this manuscript and approving it for publication was Yang Liu.

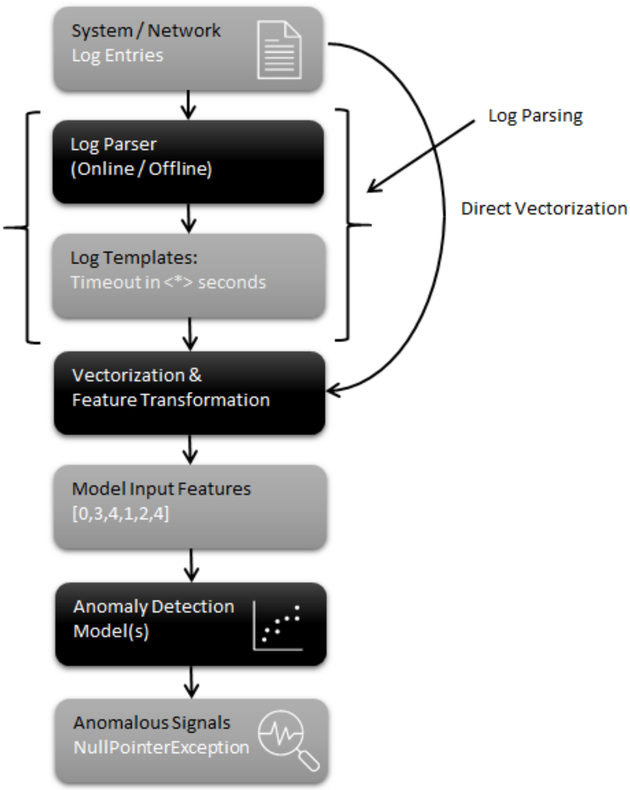

FIGURE 1. Typical log anomaly detection workflow.

format, and must be processed incrementally. The challenge of managing this unstructured data is a key focal area of log anomaly detection research.

Log anomaly detection is not a one-step process. It involves data mining to extract structured, meaningful information from raw log sources, feature transformation or vectorization to translate information into usable model input features, and anomaly detection mechanisms to detect and report anomalous signals (Fig. 1). Log anomaly detection workflows can also incorporate other functions such as preprocessing and feedback.

Although log parsers have many forms, they have the same goal: to extract log templates (also known as log signatures or events) from raw log data. Log templates represent multiple log entries of the same event, which differ only in their parameters. They are created by replacing the dynamic parameters with wildcards or placeholders.

Software generates log entries through the invocation of logging commands. Log parsers contribute to log anomaly detection by encoding unique event sequences from logs into inputs for anomaly detection models. They provide an invocation record of calls across active code branches, representing the logical execution flow of monitored processes. As parsers target the characteristics of logs, they can be considered domain-specific. Log parsers have been shown to improve the quality of generated log representations and increase downstream model performance [7]. They can prove advantageous over generic encoding methods that do not consider logging practices.

Figure 2 illustrates an example of a parser utilized within a log anomaly detection workflow. The log signatures extracted by the parser are used to vectorize a sequence of events that follow the flow of raw log entries. This sequence is encoded into an event count matrix using sliding or fixed windows. The log anomaly detection algorithm uses the event count matrix as input. With this workflow, anomalous signals can be detected from the representative numerical encoding produced by the parser and vectorization process.

Offline parsers extract templates either by directly referencing log output statements from system source code [8] or by deriving templates from historic log data through algorithmic means [9], [10], [11]. In contrast, online methods derive templates incrementally from real-time log data [12], [13], [14]. They "process log data item by item in a streaming manner, and do not require a batch of data to be available before executing" [15]. Online parsers are useful because they can be applied without source code access, historical log data, or offline training. They can be used to manage log drift through incremental template learning, and they perform the same or better than their offline counterparts in terms of average parsing accuracy [16]. This study focuses specifically on online parsers for this reason.

Like log parsing, anomaly detection methods can function online or offline. There are many forms of these methods, including statistical, machine learning, and deep learning approaches. Anomaly detection workflows can utilize individual models or ensembles. They typically target abnormal log events or abnormal sequences of events. They can also target abnormal parameter sequences, timing abnormalities, and other combinations of such features. DeepLog, for example, uses multiple LSTM models to target log events and parameter sequences with timing-related metadata integrated into the feature set [17]. However, the performance of log anomaly detection approaches varies widely. This performance variance is apparent not only between methods but also depending on the log source analyzed (see Section VI).

There have been several surveys on log anomaly detection-related topics over the past years. These generally have focused specifically on parsing technologies or particular types of anomaly detection (such as deep learning) independent of the types of parsers being used [18], [19]. To our knowledge, this is the first survey focusing on the intersection of online parsers and anomaly detection methods. It makes the following contributions. First, we summarize and consolidate the current state of research into online parser-supported log anomaly detection workflows, taking inventory of all relevant components. Second, we analyze recent trends in research, including the evolution of achievable accuracy for these components. Third, we propose a new taxonomy to describe and categorize these workflows, summarizing all relevant studies to date using this taxonomy. Finally, we highlight and discuss research gaps discovered through our analysis and provide direction for future research.

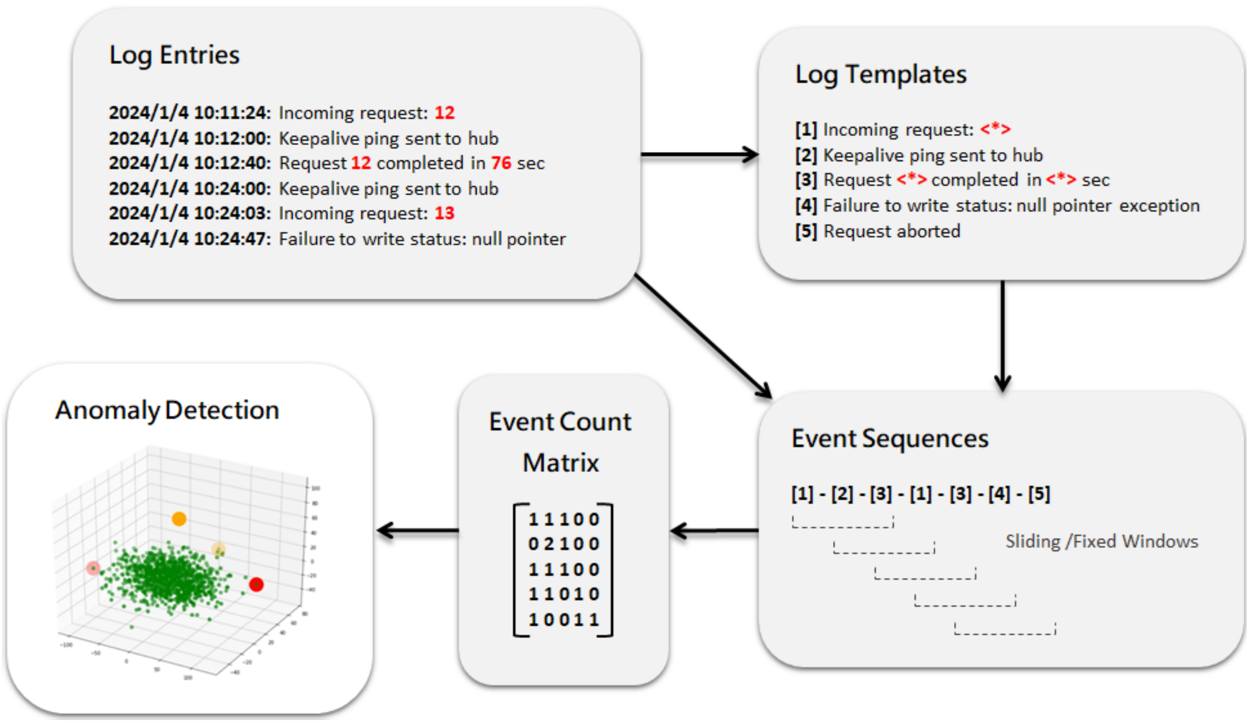

FIGURE 2. Example of a log anomaly detection method using log parsing.

Ultimately, the motivation for this study is to support the industrialization of online log anomaly detection methods for production system monitoring. Through this exploration of the current landscape of online parser-supported log anomaly detection research, we hope to facilitate this process.

The rest of this paper is organized as follows. Section II describes the five research questions addressed in this study. Section III explains the research method used to formulate the responses to these questions, including a description of our systematic literature review process and how method performances were compared. Section IV addresses Research Question (**RQ**) **1**, examining log anomaly detection research trends. Section V addresses **RQ2**, comparing online parsing method performances and presenting parser performance trends over time. Section VI formulates a response to **RQ3**, performing a similar comparison of online parser-supported log anomaly detection method performances. Section VII addresses **RQ4**, presenting a taxonomy to classify online parser-supported log anomaly detection workflows. Section VIII focuses on **RQ5**, identifying and discussing the research gaps discovered through our study. Section IX reviews related works. Section X discusses internal and external threats to validity. Finally, Section XI presents our conclusions, including a discussion of future research directions.

II. RESEARCH QUESTIONS

We aim to assess the state of online parser-supported log anomaly detection research by addressing the following research questions:

RQ1. What are the current research trends in onlineparser-supported log anomaly detection research? To answer this question, we compared citation counts for online parsers discovered through our previous systematic literature reviews to those since 2021 (extracted from Scopus[1]) [20], [21]. We divided these recent studies by type and compiled statistics on all newly introduced methods. Using this data, we performed a trend analysis.

RQ2. How has the performance of online log parsers evolved? To answer this question, we compiled an up-to-date inventory of online parsing methods using our systematic literature reviews. We compiled evaluation results using the most common public datasets and metrics found across studies and graphed them in order of method introduction date.

RQ3. How has the performance of online parser-supported log anomaly detection methods evolved? To answer this question, we combined online parser-supported log anomaly detection methods discovered in our previous paper with those found in this study's systematic literature review [21]. We then compiled reported metrics for these methods using the most frequently used public log datasets.

RQ4. How can different forms of online log anomaly detection be classified? To answer this question, we developed a taxonomy based on the log anomaly

[1]https://www.scopus.com

TABLE 1. Online/incremental log parser citations.

Parser	Publications	Total Citations	Since 2021
Drain	2017 [14]	**358**	**292**
Spell	2016 [22] / 2019 [13]	204 / 43	141 / 37
SHISO	2013 [12]	79	53
SwissLog	2020 [23] / 2023 [24]	49 / 0	49 / 0
FT-tree	2017 [25]	68	47
Logram	2020 [26]	29	29
LogSimilarity	2015 [27] / 2019 [28]	47 / 11	15 / 10
Paddy	2020 [29]	17	17
Craftsman[2]	2020 [30]	19	16
Logan	2019 [31]	18	16
LogOHC	2019 [32]	15	11
OILog	2021 [33]	10	10
FLP	2018 [34]	5	5
LTmatch	2021 [35]	5	5
One-to-one	2020 [36]	5	5
BSG	2018 [37]	4	3
OLMPT	2020 [38]	3	3
Slop	2018 [39]	2	2
LenMa	2016 [40]	0	0

detection methods discovered through our studies. We then verified the taxonomy by using it to classify these methods.

RQ5. Does existing online parser-supported log anomaly detection research contain gaps that merit future exploration? To answer this question, we assessed the studies discovered through our systematic literature reviews, compiled potential issues, and highlighted areas that we found to be lacking in coverage. We discuss the significance of these gaps and the potential for future research to improve upon these areas.

This paper extends our preliminary research results presented at APSEC 2021 as part of the ERA (Early Research Achievements) track [21]. All research questions presented in this study are extensions of the original literature review. All figures, data, and conclusions drawn from the original work are cited accordingly.

III. RESEARCH METHOD

In this study, we performed a refreshed literature review of online parser-supported log anomaly detection. We compiled and compared the results from evaluations discovered through this review to perform method comparisons. This process provided the foundation for this study and is described below in greater detail.

A. SYSTEMATIC LITERATURE REVIEW

To address the RQs, we initiated a refreshed systematic literature review of online parser-supported log anomaly detection methods using the results from our two previous studies. Our first study yielded 358 results for a keyword search of "log parsing" via Scopus [20]. After excluding articles not written in English and duplicates, 340 studies remained. These articles were reviewed, irrelevant articles were discarded, and research targeting online/incremental approaches to log parsing were selected. Snowballing was performed using citation searches in Research Gate,[3] and a final list of online parsers was compiled.

In our subsequent ERA (Early Research Achievements) publication, we performed a systematic literature review of online parser-supported log anomaly detection methods using citations of online parsers discovered through our previous study. A search in Scopus resulted in 276 articles. Of these, 124 were duplicates or written in a non-English language [21]. Of the remaining 152, relevant log anomaly detection methods were compiled, and the results were summarized and presented for discussion.

This research used our previous survey results to initiate an up-to-date review of online parsers and online parser-supported log anomaly detection methods. We performed a citation search in Scopus for all previously discovered studies to extract a collection of new relevant literature (reflecting the data available as of January 1st, 2024). We analyzed modern research trends by comparing statistics on recent studies with those from our previous literature reviews (**RQ1**). We used this refreshed review to compile and compare online parser and online parser-supported log anomaly detection method performance (**RQ2-RQ3**). We developed a taxonomy of online parser-supported log anomaly detection methods and verified it by classifying all methods discovered through our literature reviews (**RQ4**). Finally, we summarized existing research gaps to suggest directions for future work (**RQ5**)

B. PERFORMANCE COMPARISONS

To compare the performance of online parsers and online parser-supported log anomaly detection methods, we compiled the results of evaluations from studies identified through our current and previous systematic literature reviews [20], [21]. When compiling the results, we prioritized method evaluations performed within their own introductory papers. Any scores that deviated heavily from those discovered in other comparative studies were discarded. We chose evaluations using the most frequently utilized public log datasets and standard metrics for the broadest comparison possible. Performance was graphed in the order of method introduction. We analyze and discuss performance trends using this data in Sections V and VI.

IV. LOG ANOMALY DETECTION RESEARCH TRENDS

In previous work, we compiled all known online parsing and online parser-supported log anomaly detection methods [20], [21]. To analyze modern log anomaly detection research

[2] Newly discovered from a survey in this paper's literature review [41].

[3] https://www.researchgate.net

TABLE 2. Breakdown of recent log anomaly detection studies (2021 - January 1st, 2024).

Parsing Type	Parsing Method	Machine Learning Model Usage	Deep Learning Model Usage	**Total**
Online Parsing	Drain	[7] [42] [43] [44] [45] [46] [47] [55] [56] [57] [58] [59] [60]	[7] [45] [47] [48] [49] [50] [51] [52] [53] [54] [57] [60] [61] [62] [63] [64] [65] [66] [67] [68] [69] [70] [71] [72] [73] [74] [75] [76] [77] [78] [79] [80] [81] [82] [83] [84] [85] [86] [87] [88] [89] [90] [91] [92] [93] [94] [95] [96] [97] [98] [99] [100] [101] [102] [103] [104] [105] [106] [107] [108] [109][4]	**69**
	Drain3[5]	[110] [111]	[111]	**2**
	FT-tree	[112] [113]	[69] [114] [115]	**5**
	Spell	[116]	[69] [73] [98] [117] [118] [119] [120] [121] [122] [123]	**11**
	TCN-Log2Vec		[124]	**1**
Offline Parsing	ADAL-NN		[125]	**1**
	MDFULog		[126]	**1**
	FastLogSim		[127]	**1**
	GAN-EDC		[128]	**1**
	iPLoM		[69]	**1**
	LKE		[69]	**1**
	Logsig		[69]	**1**
	LPV		[129]	**1**
	Polo		[130]	**1**
Other	Custom/Manual	[131] [132] [133] [134] [135]	[4] [133] [136] [137] [138] [139] [140] [141]	**12**
	From Source		[142][6]	**1**
	Unspecified	[143] [144] [145] [146] [147] [148]	[146] [148] [149] [150] [151] [152] [153] [154] [155] [156] [157] [158] [159] [160] [161]	**19**
	No Parser[7]	[7] [162] [163] [164] [165] [166]	[3] [7] [61] [88] [98] [166] [167] [168] [169] [170] [171] [172] [173] [174] [175] [176] [177] [178] [179] [180] [181] [182] [183] [184] [185] [186] [187]	**31**
	Total	**34**	**125**	

trends, we refreshed this review through the compilation of research conducted since 2021 using citation searches in Scopus. This search yielded 766 new citations (Table 1). These citations were filtered to remove duplicates and non-English language articles. The resulting parser and log anomaly detection method studies with relevance are listed in Tables 2 and 3.

[4]Uses a BiGAN with an ensemble of "base classifiers" [109].
[5]https://github.com/IBM/Drain3
[6]Uses Drain as part of online template matching.
[7]Includes direct vectorization with or without regex style filtering.

As seen from Table 1, since 2021, the most frequently cited online parsers continue to be Drain (38%) and Spell (23%). In contrast, parsers with equal or higher average PA scores using the LogHub public log data collection [143], [188] have been cited significantly less (see Fig. 4 and 8). Parsers such as LTmatch [35], Paddy [29], and SwissLog [23], for example, have higher average PA values reported using the 16 log datasets in LogPAI's Loghub. Still, they represent only a tiny percentage of the online parser citations since 2021 (1%, 2%, and 6% respectively). This under-representation highlights a gap in modern log anomaly detection research that merits future attention (discussed further in Section VIII).

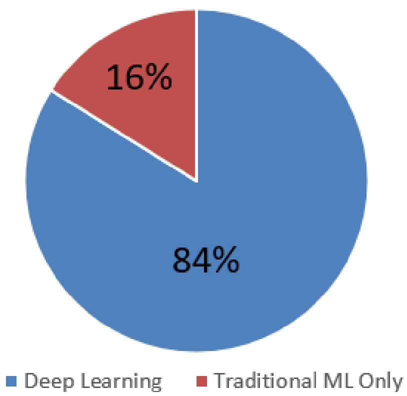

FIGURE 3. Deep learning usage for log anomaly detection since 2021.

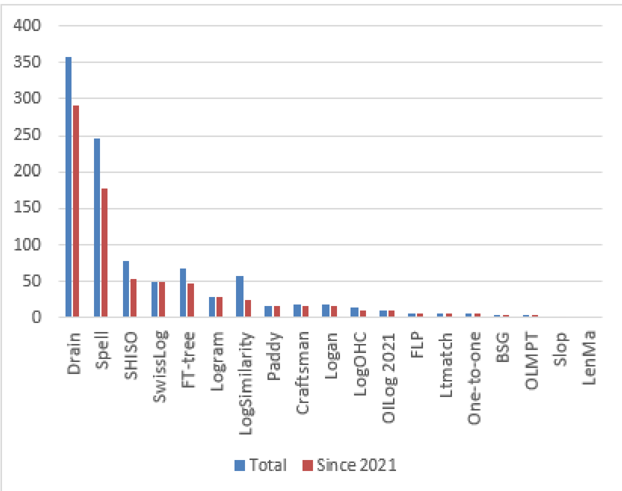

FIGURE 4. Online parser study citations totals.

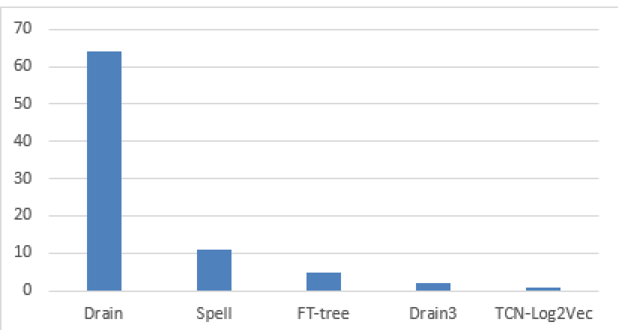

FIGURE 5. Online parser use in studies since 2021.

A. DEEP LEARNING UTILIZATION

Of the 149 new log anomaly detection studies discovered since 2021, the vast majority (84%) use deep learning approaches, signifying a continued shift from more traditional anomaly detection techniques (Fig. 3). Over half (56%) used online parsing methods, 86% of which were used in combination with deep learning (Fig. 7 and Table 2).

TABLE 3. Parsing methods introduced since 2021.

Parser	Publication	Online	Parallel
AdaptParse	2023 [189]	No	No
ADAL-NN	2023 [125]	No	No
ADR	2021 [190]	No	No
Bertalan and Alois	2023 [191]	No	No
Biglog	2023 [88]	No	No
Brain	2023 [192]	Yes	No
ChatGPT	2023 [193]	No	No
Cognition	2023 [194]	Yes	No
DIP	2022 [195]	No	No
Drain+	2022 [196]	Yes	No
Drain3	2021 [110]	Yes	No
eLP	2022 [197]	Yes	No
Fuzzy Mining	2021 [198]	No	No
GAN-EDC	2021 [128]	No	No
Hue	2023 [199]	Yes	No
LFP	2021 [200]	No	Yes
Log3T	2023 [201]	No	No
LogDTL	2021 [202]	No	No
LogPPT	2023 [203]	No	No
LogPunk	2021 [204]	Yes	No
LogSlaw	2023 [205]	Yes	No
LogStamp	2022 [206]	No[8]	No
LTD-MO	2022 [207]	No	No
LTmatch	2021 [35]	Yes	No
Marlaithong *et al.*	2023 [208]	No	No
MDFULog	2023 [126]	No	No
ML-Parser	2021 [209]	Yes	No
OILog	2021 [33]	Yes	No
PatCluster	2023 [210]	No	No
PC	2022 [211]	No	No
PILAR	2023 [212]	No	No
Polo	2023 [130]	No	Yes
Prefix-Graph	2021 [213]	Yes	Yes[9]
PVE	2023 [214]	No	No
QuickLogS	2021 [215]	No	No
Semlog	2023 [216]	No	No
SNNLog	2023 [217]	No	No
Spell+	2021 [204]	Yes	No
SPINE	2022 [218]	No	Yes
Spray	2022 [219]	Yes	No
TCN-Log2Vec	2023 [124]	Yes	No
ULP	2022 [220]	No	No
UniParser	2022 [221]	No	No
USTEP	2021 [222] / 2023 [223]	Yes	No
USTEP-UP	2021 [222] / 2023 [223]	Yes	Yes
VALB	2023 [224]	No	No

In comparison, 21% used direct vectorization or NLP (omitting template extraction via parsing), 8% utilized custom or manual parsing, and only 5% utilized offline parsing. All studies using offline parsing also used deep learning anomaly detection. The remaining 19 studies used some form of a parser, but the details were omitted.

B. PARSING METHOD UTILIZATION

Figure 5 shows that Drain continues to be the most commonly used parser since 2021 followed distantly by Spell. Drain was used in 82% of online parser-supported studies and Spell was used in 13% (Table 2). Only seven studies used offline

[8]Described as online but doesn't include incremental learning.
[9]Can be implemented in parallel mode.

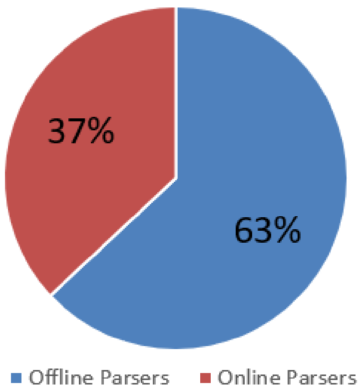

FIGURE 6. Distribution of online/offline parsers introduced since 2021.

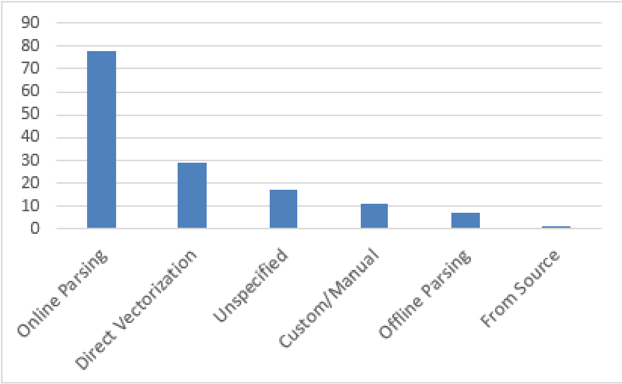

FIGURE 7. Parsing types used in log anomaly detection studies since 2021.

parsing, illustrating that such methods have drastically fallen out of favor.

Numerous parsing methods have been introduced since 2021, demonstrating that log parser research remains extremely active (Table 3). 37% of the methods introduced are online (Fig. 6). Three are derivatives of Drain and Spell. Modern log anomaly detection studies primarily use online parsing (see Fig. 7), so the volume of new offline parsers introduced (i.e., 29 methods) is somewhat surprising. Of the 46 new parsers discovered in total, only five support parallel processing (Table 3).

RQ1. What are the current research trends in onlineparser-supported log anomaly detection research? **Log parsing continues to be an extremely active research area. Since 2021, 46 new parsing methods have been introduced (Table 3). 37% are online methods, and 63% are offline (Fig. 6). As over half of the log anomaly detection studies since 2021 used online parsing, the comparatively large number of newly introduced offline parsers is surprising (Fig. 6 and 7). Drain, followed by Spell, are the most commonly utilized parsers (Fig. 5). Most studies used deep learning techniques, demonstrating a shift away from traditional machine learning and statistical algorithms (Fig. 3). Although direct vectorization methods are becoming more common (21%), online parsing workflows remain the most popular overall (Fig. 7).**

V. ONLINE LOG PARSER PERFORMANCE

Our systematic literature reviews discovered 33 online parsing methods in total. Three (i.e., Drain3, Drain+, and Spell+) are derivative implementations of preexisting approaches, and 48% (39% excluding derivatives) were introduced since 2021. These methods are listed in Table 4.

We assess the performance of online parsers by comparing the results of studies discovered through our systematic literature reviews. We compile the results from the most commonly utilized datasets and metrics to perform this comparison. These results provide an inventory of available online parsers and a reference for their performance.

A. AVERAGE PA

Parsing accuracy (PA) is a metric representing the ratio of correctly parsed log entries relative to the total number of entries evaluated [16]. It is a standard metric that can be used for comparing different parsing methods. Figure 8 shows that online log parser performance has steadily increased since the original introduction of SHISO [12] in 2013. PA values using the 16 log datasets in LogHub (representing both parser accuracy and robustness) have gradually improved. A significant portion of this improvement (0.751 to 0.865) coincides with the introduction of the Drain parser in 2017. This improvement may be why Drain remains the most heavily utilized online parser in log anomaly detection research (Fig. 5).

The average PA achievable against the 16 log datasets in LogHub has improved with the introduction of recent parsers such as Paddy [29], SwissLog [24], LTmatch [35], LogPunk [204], Drain+ [196], Hue [199], and Brain [192]. However, the improvement margin has decreased due to the higher overall level of accuracy demonstrated by modern methods in general. Experimentation with these modern parsers in anomaly detection workflows would still be worthwhile. Their lack of representation in log anomaly detection studies is a significant research gap, and this topic is discussed in more detail in Section VIII.

B. OTHER PERFORMANCE METRICS

Aside from PA, other commonly used parser performance metrics include precision, recall, F-score, and the Rand index. This study compiles available evaluation results using these metrics to provide a broad performance comparison. Although Dendrogram purity [32], Levenshtein edit distance [225], and loss functions [31] also appear, they are used infrequently, and thus excluded from our summary.

Several studies use a stricter form of PA requiring all dynamic parameters to be identified for a template to be considered correctly parsed [26], [194]. This form of PA is used in only a limited number of studies, and like with the metrics previously mentioned, we have excluded it for

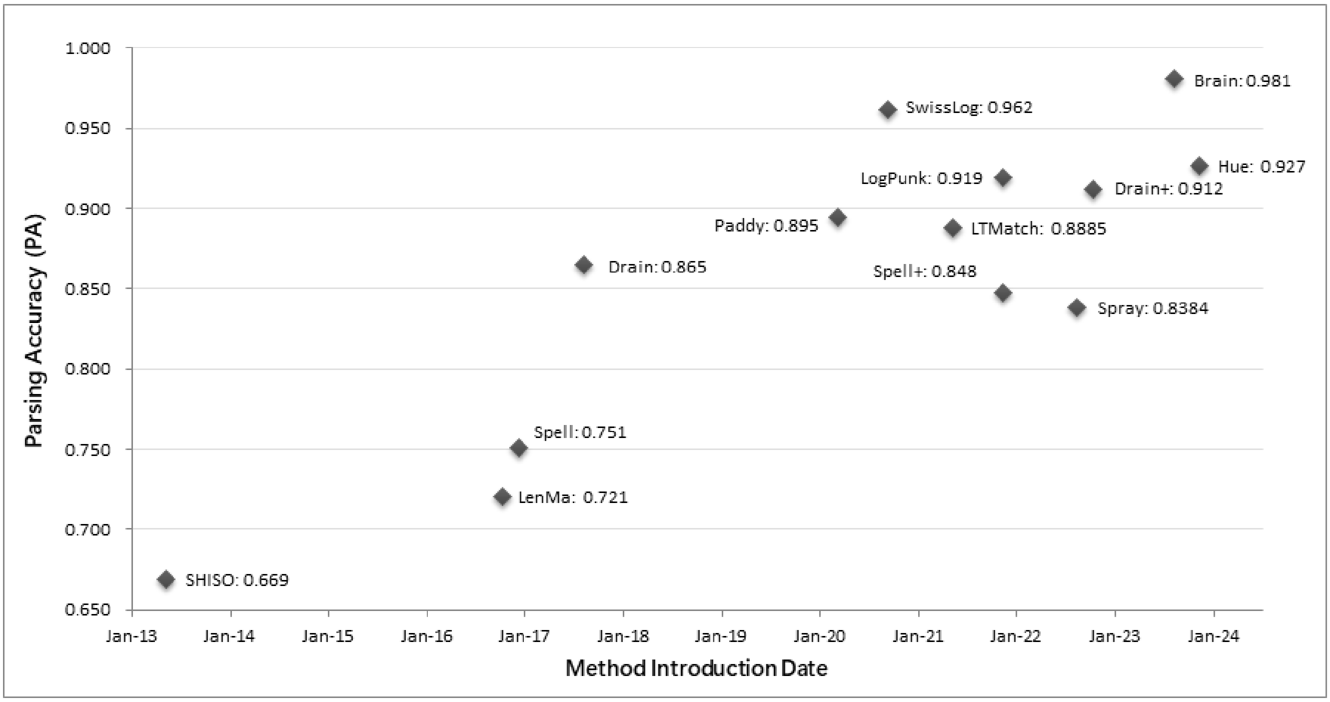

FIGURE 8. Average PA of online parsing method evaluations using the 16 public log datasets in LogPAI's Loghub.

this reason. However, it would provide for a higher-quality assessment if its use was more widely adopted.

Table 4 summarizes the reported PA, F-score, and Rand index values achieved for the most common public log datasets used in online parser comparison studies (i.e., the HDFS and BGL log datasets). Parsers tend to perform extremely well against the HDFS dataset with a minimal score deferential. One reason for these high scores is the low diversity of log statement formats. With over 11 million log entries, the HDFS dataset contains only 30 unique templates (14 from the 2k entry sample provided by LogHub) [188]. This issue is discussed in more detail in Section VIII.

The BGL dataset has relatively more templates (619 from over 4.5 million log entries), making gaps in parser performance more apparent. With this dataset, it can be seen that newer parsers such as Brain, LogPunk, Paddy, and SwissLog match or outperform the Drain parser in terms of PA. Note that these results are also reflected in the methods' average PA values recorded against the 16 datasets in LogPAI's Loghub (Fig. 8).

In regards to Rand index values, Prefix-Graph outscores Drain for the BGL dataset (0.993 versus 0.912), and Drain outperforms Prefix-Graph for the HDFS dataset (1.000 versus 0.989). However, Prefix-Graph matches or outperforms the Drain parser on seven of the ten datasets evaluated in its study (with a higher average Rand index value of 0.975 versus Drain's 0.953) [213]. It also matches or outperforms Spell and FT-tree on eight of these datasets.

RQ2. How has the performance of online log parsers evolved? **Since 2021, 17 new online parsers have been introduced, three being derivative implementations of previous methods. Table 4 lists all known online parsers and their PA, F-score, and Rand index values achieved against the BGL and HDFS public log datasets. The performance of online parsers has gradually increased over time (Fig. 8). Modern parsing methods score very high in accuracy and robustness. Although Brain [192] shows the highest recorded average PA for the 16 public log datasets in LogPAI's Loghub, it hasn't been used in log anomaly detection research (Table 2). In contrast, Drain and Spell remain heavily utilized, even with their lower average PA scores.**

VI. LOG ANOMALY DETECTION PERFORMANCE

Log parsers have been generally well assessed for robustness through the use of many public log datasets. Log anomaly detection methods, however, have not benefited from the same level of evaluative coverage. These studies generally utilize only a small number of datasets for evaluation. Out of those used, the HDFS and BGL datasets are the most common. To perform a broad performance comparison, we utilize these same datasets with common metrics. Performance results were ordered by the date of anomaly detection method introduction, and we analyzed the evolution of performance improvements seen over time. The results of this analysis are discussed below in the following sections.

TABLE 4. Online/incremental log parser performance (BGL/HDFS Datasets).

Parser	BGL PA	BGL F1	BGL RI	HDFS PA	HDFS F1	HDFS RI
Brain	**.998**			.998		
BSG		.99			**1**	
Cognition		.9992			**1**	
Craftsman						
Drain	.963	.9996	.912	.998	**1**	**1**
Drain3						
Drain+	.941	.999		**1**	**1**	
eLP						
FLP		.999			**1**	
FT-tree			.91			.935
Hue	.849	.700		.998	.867	
LenMa	.69			.998		
Logan						
LogOHC					**1**	
LogPunk	.979			.998		
Logram						
LogSimilarity						
LogSlaw		.99			.99	
LTmatch	.9325			**1**		
ML-Parser	.81			.85		
OILog					**1**	
OLMPT		**1**			**1**	
One-to-one	.9610	.9996		**1**	**1**	
Paddy	.963			.940		
Prefix-Graph			**.993**			.989
SHISO	.711	.87		.998	.93	
Slop		.94			.93	
Spell	.787	.98	.88	**1**	**1**	.999
Spell+	.822			.998		
Spray	.8655			.9985		
SwissLog	.97			**1**		
TCN-Log2Vec	.94			.99		
USTEP	.964			.998		

A. HDFS DATASET ASSESSMENT

The first log anomaly detection method evaluated using an online parser discovered through our systematic literature review was the PCA algorithm, used in the introductory paper for the Drain parser in 2017 [14]. This paper compares the performance of different offline and online parsers (Drain, SHISO, Spell, and IPLoM) used in combination with PCA as part of a log anomaly detection workflow. Although the F-score values for this study were not directly reported, we were able to calculate them using the metrics presented in the paper in combination with known features of the HDFS dataset. In this study, Drain (online) and IPLoM (offline) had the highest overall performance. Used with PCA, they both produced an F-score value of 77.02%. With Spell and SHISO, this value dropped to 76.83% and 74.57% respectively.

Zhang et al. evaluated their semi-supervised and unsupervised anomaly detection methods using a similar comparison of the Drain, AEL, and IPLoM parsers [44]. The F-score values against the datasets in their study increased when using Drain in combination with their semi-supervised method (sADR). Using their unsupervised anomaly detection method (uADR), Drain outperformed the other approaches in half of the cases. This illustrates that parser choice can significantly impact log anomaly detection workflow performance.

Several months after the original Drain parser study in 2017, the DeepLog anomaly detection method was introduced. This method, which utilizes online parsing and a parallel LSTM deep learning approach, was evaluated against the HDFS dataset [17]. DeepLog significantly improves performance over PCA with an F-score of 96%. Since then, improvements have continued. Many new online parser-supported log anomaly detection methods have been introduced with higher reported scores (Fig. 9). LCC-HGLog and Zhang et al. have achieved the highest recorded F-score against this dataset (99.9%) [71], [105]. Many other methods have realized F-scores above 99%, starting with LogRobust in 2019 [226]. Although the vast majority of studies have yielded F-scores over 95%, several have failed to do so [57], [76], [86], [104], [106], [107], [227]. For viewing ease, Figure 9 omits these studies. These methods, however, are included in our analysis in subsequent sections.

B. BGL DATASET ASSESSMENT

Figure 10 summarizes the log anomaly detection method F-score values recorded against the BGL dataset. Most have

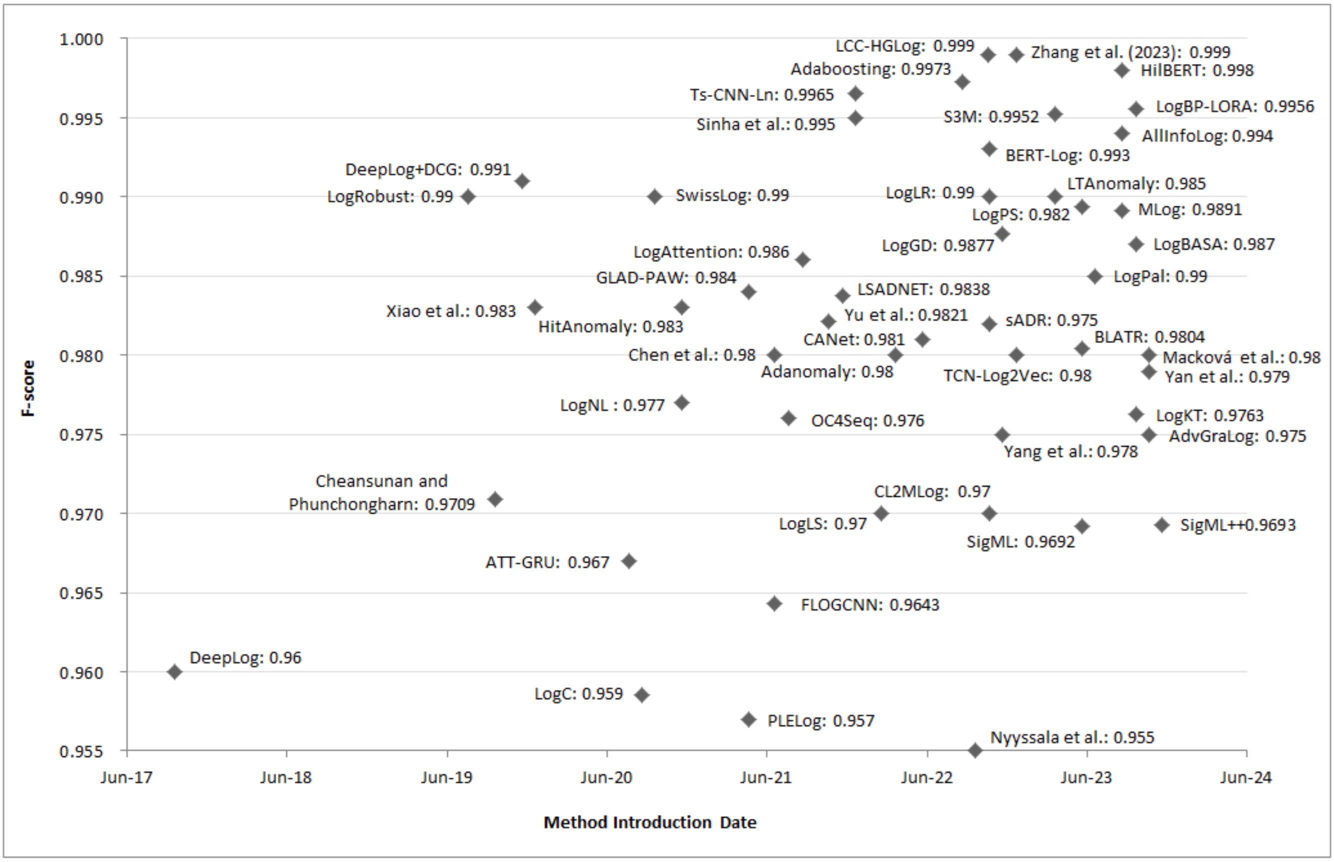

FIGURE 9. Evolution of online anomaly detection method performance (F-score) using the public HDFS dataset.

TABLE 5. Rule-based failure detection accuracy.

Dataset	Precision	Recall	Accuracy	F-measure
BGL	.3306	.7332	.8714	.4557
HDFS	.0469	.3846	.7556	.0837

achieved scores at or above 90%. Those that have performed worse have been excluded for readability. This high level of performance is significant given the relative complexity of the dataset. It illustrates the strength of recent online parser-supported log anomaly detection methods against complex log targets.

Figure 11 shows a more detailed view of these results. Many of the methods recorded higher recall values than precision. This suggests that although these methods may be proficient at detecting anomalies, they likely produce many false positives. False positives are a significant concern for method industrialization since these signals can drown out true alerts when used for systems monitoring. To deal with such issues, false positive mitigation strategies such as model feedback mechanisms are required. Table 6 (discussed in more detail in subsequent sections) confirms that these mechanisms have yet to be adequately researched. We discuss this topic in more detail in Section VIII-B.

C. PERFORMANCE AGAINST BOTH DATASETS

Many recent online parser-supported log anomaly detection methods have performed well against both the HDFS and BGL datasets. SwissLog, for example, has an F-score of 99% recorded against both. Figure 8 shows that the SwissLog parser performs the second highest in terms of average PA for the sixteen public log datasets in LogPAI's Loghub. This high average parsing accuracy is likely a supporting factor for the method's success against multiple datasets.

Like SwissLog, LCC-HGLog, LogBP-LORA, AllInfoLog, Zhang et al. (2023), BERT-Log, LogLR, LogPal, and S3M are also robust against both the HDFS and BGL datasets, achieving F-score values at or above 99% (Fig. 9 and 10). A common factor amongst these methods is their use of deep learning. They also all use either semantic sequencing or graphical feature encoding.

D. RULE-BASED METHOD COMPARISON

The use of rule-based methods for log monitoring in industrial settings is ubiquitous. These methods trigger

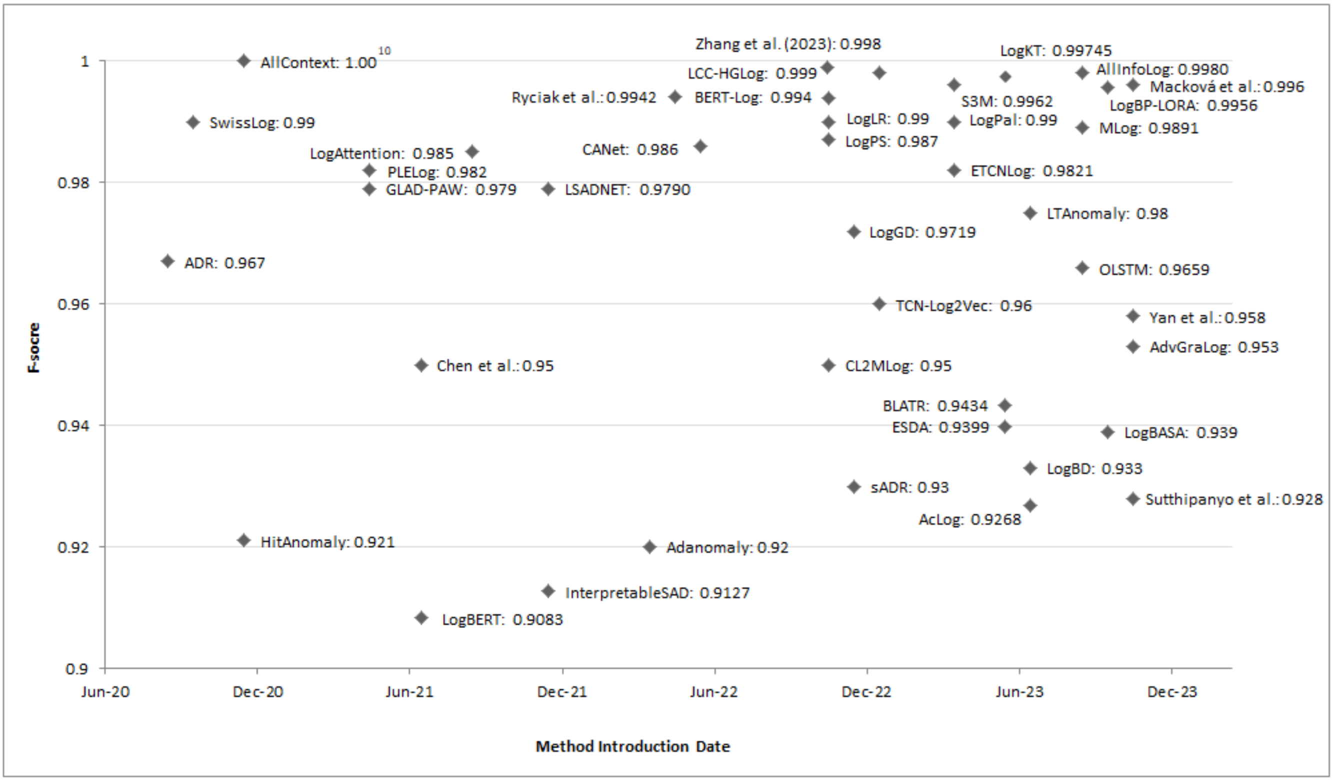

FIGURE 10. Evolution of online anomaly detection method performance (F-score) using the public BGL dataset.

alerts based on rules configured to detect the presence of specified keywords in log entries. Crafting rules capable of detecting unforeseen issues, however, is challenging. The manual creation and maintenance of rules is also very time-consuming. Production service engineers often rely upon a standard configuration set for this reason.

In our previous work, we compared rule-based methods to anomaly detection approaches using an industrial dataset [228]. We found that anomaly detection methods were much more accurate, but suffered from a large number of false positives. In this study, we extended the *Evaluator* class of our component-based log anomaly detection pipeline framework to support the assessment of the BGL and HDFS public log datasets. We then assessed the pipeline's rule engine component with these datasets, measuring its performance using a standard set of industry keyword rules (containing the tokens "error," "exception," and "failure").

The results of this experiment are shown in Table 5. As can be seen, the online parser-supported log anomaly detection methods summarized in this review (Fig. 9 and 10) significantly outperform the rule-based approach. The rule-based approach also resulted in an extremely large number of false positives (131,462 from the HDFS dataset and 517,401 from the BGL dataset).

[10]Uses a filtering algorithm to reduce related sets of alerts to a single initial alert per failure [229], [230].

RQ3. How has the performance of online parser-supported log anomaly detection methods evolved? **The achievable accuracy of online parser-supported log anomaly detection methods (as reported through the use of F-score values against the HDFS and BGL public log datasets) has steadily increased over time. However, many of these methods produce comparatively high recall values for the BGL dataset, suggesting the presence of a large number of false positives. These methods are generally effective against the log types used and hold promise for real-world adaptive system monitoring tasks. They also perform significantly better than traditional rule-based approaches.**

VII. TAXONOMY

Online parser-supported log anomaly detection methods are constructed from a composition of components (both mandatory and optional). Here, we classify these methods using the types of components utilized. First, we construct a taxonomy of online parser-supported log anomaly detection workflows based on their components (Fig. 13). We then verify the taxonomy by using it to categorize the online parser-supported log anomaly detection methods discovered through this study. This taxonomy serves to organize modern research into core functional categories, elucidate component attributes and features, and highlight coverage gaps to inform future studies.

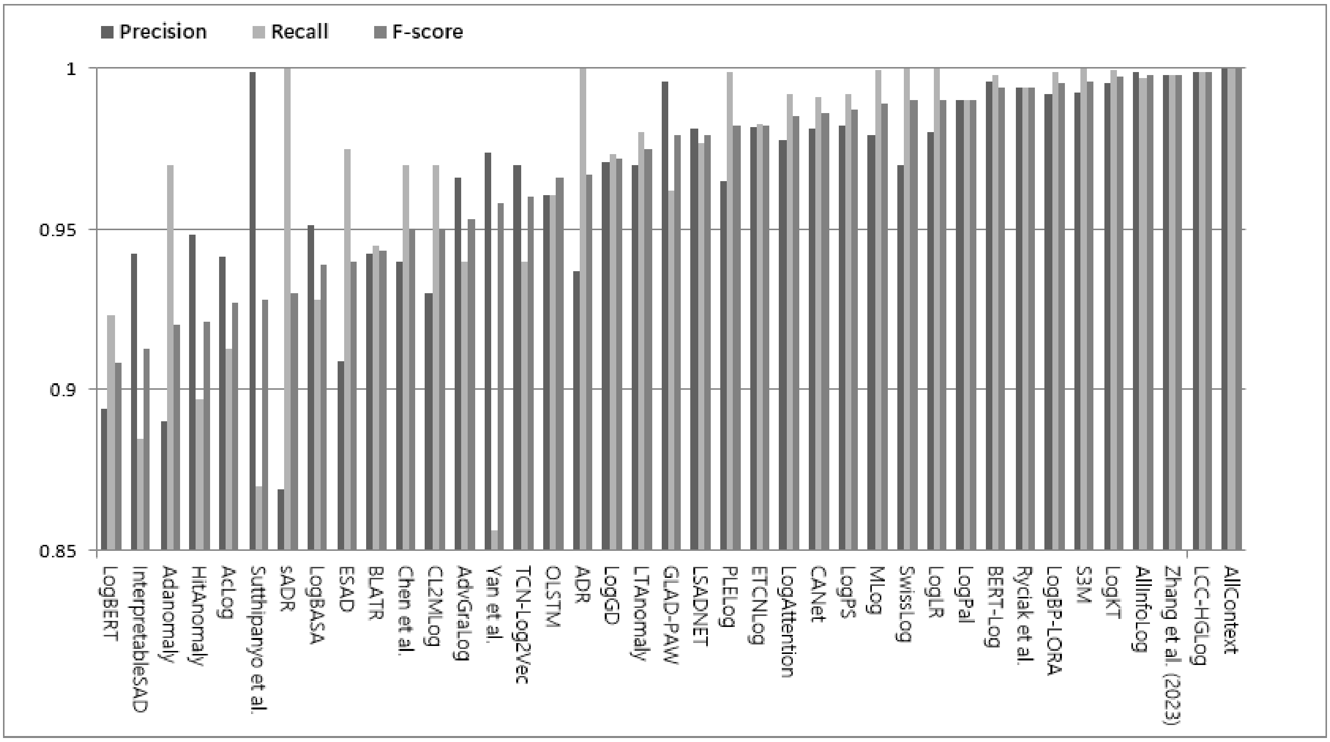

FIGURE 11. Online anomaly detection method performance (precision, recall, and F-score) using the BGL dataset.

A. PRIMARY COMPONENTS

1) PREPROCESSING

Some log anomaly detection workflows utilize preprocessing components. These components include NLP functions and filters to remove characters and character sequences (such as punctuation or stop-words) [174]. Another tactic is term-splitting, which aims to separate connected tokens (for example, splitting the string ''TimeoutException'' into ''Timeout'' and ''Exception'') [229]. Another approach is replacing predetermined character patterns with wildcard symbols (e.g., IP addresses with "<:IP:>"). These domain-specific replacement rules have been shown to improve parser performance, and some parsers even include token replacement functions as a data-cleansing step [14], [231].

2) PARSING

Log parsing is an active area of research with new techniques being introduced continuously (Table 3). Methods such as LenMa, ML-Parser, and SwissLog use clustering [23], [40], [209]. Parsers based on heuristics are also popular, having been shown to work well with many different anomaly detection methods [232]. Heuristic techniques are frequently coupled with fixed-depth parsing trees, as seen with Drain, FT-tree, Hue, OLMPT, and TCN-Log2Vec [14], [25], [38], [124], [199]. Many modern parsers such as Brain, Cognition, and Craftsman use parsing trees with variable depth [30], [192], [194].

Longest Common Subsequence (LCS) is an algorithm used for log parsers based on the observation that ''the constant representing a message type often takes most of the sequence and the parameter values assume only a small portion'' [22]. However, this approach alone can lead to under-partitioning [14]. Frequent Pattern Mining (FPM) is a well-known parsing approach utilized for offline parsers such as SLCT, as well as several online parsers (Fig. 12) [233].

Of the more recent methods, newer techniques such as Evolving Granular Classifiers (eGC) and Keyword Extraction have been used [33], [197], [234]. Paddy employs a dynamic dictionary for parsing, and LogSlaw uses a static one [29], [205]. Prefix-Graph uses a graph representation [213]. MoLFI and LTD-MO, both offline parsers, use evolutionary and swarm optimization algorithms [11], [207]. Note these categories have been excluded from our taxonomy as no online methods were discovered that use them.

As shown in Figure 12, diverse techniques are used to implement log parsers. These techniques are used both in isolation and in combination with others. As parsing method research continues and new methods are introduced, these techniques are expected to grow and expand.

3) ENCODING

Zhao, Jiang, and Ma introduced three feature categories: *log event count vectors*, *log event index sequences*, and *log event semantic vectors* [234]. Ma et al. presented an alternative classification method consisting of the following categories:

	Clustering	Dictionary (Dynamic)	Dictonary (Static)	Evolving Granular Classifier (eGC)	Frequent Pattern Mining (FPM)	Graph	Heuristics	Longest Common Subsequence (LCS)	Parsing Tree (Fixed Depth)	Parsing Tree (Variable Depth)	Keyword Extraction
Brain							✓			✓	
BSG							✓				
Cognition							✓	✓		✓	
Craftsman					✓					✓	
Drain							✓		✓		
Drain3							✓		✓		
Drain+							✓		✓		
eLP				✓							
FLP	✓						✓				
FT-tree					✓		✓		✓		
Hue							✓		✓		
LenMa	✓										
Logan								✓			
LogOHC	✓										
LogPunk							✓				
Logram					✓						
LogSimilarity	✓						✓				
LogSlaw			✓				✓				
LTMatch								✓	✓		
ML-Parser	✓							✓			
OILog											✓
OLMPT							✓		✓		
One-to-one							✓				
Paddy		✓					✓				
Prefix-Graph						✓					
SHISO										✓	
Slop							✓	✓			
Spell								✓			
Spell+								✓			
Spray								✓			
SwissLog	✓							✓		✓	
TCN-Log2Vec							✓		✓		
USTEP										✓	

FIGURE 12. Online parser method classifications.

counts, *indexes*, *events*, *sequences*, *time*, *parameters*, *graphical features*, and *others* [235]. Our systematic literature review confirmed these latter categories to be comprehensive, and we have included them as-is within our log anomaly detection workflow taxonomy.

Note that with this categorization method, sequence features imply the use of event windows, but the type of window (i.e., fixed, sliding, or session-based) is not determinable. The encoding strategies for event windows, however, are highly dependent on the logs being assessed. Session windows can be used and are oftentimes preferred when a session identifier is available (as with the HDFS dataset block ID for example). Fixed or sliding windows are generally selected when these identifiers are not available (as with the ThunderBird dataset) [236]. The window type is therefore less of a feature of the log anomaly detection method and more of an adaptation based on the log source. For this reason, we feel it is reasonable to omit them as distinct encoding types within the taxonomy.

4) ANOMALY DETECTION

Log anomaly detection methods come in many forms. These can be divided into statistical, traditional machine learning, and deep learning types. In modern online parser-supported log anomaly detection research, deep learning is the most heavily utilized (Fig. 3).

Table 6 classifies the log anomaly detection methods discovered through our systematic literature review. The majority of these studies (84%) use deep learning. Utilized methods include neural networks (NN), different forms of recurrent neural networks (RNN), graph neural networks (GNN), convolutional neural networks (CNN), generative adversarial networks (GAN), transformers, autoencoders (AE), and logical tensor networks (LTN) [237]. The remaining 16% use statistical and traditional machine learning approaches. These include supervised, unsupervised, and dimensionality reduction methods. In some cases, ensembles of multiple model types are used as well.

5) FEEDBACK

Feedback mechanisms provide corrective information back to log anomaly detection models to enhance performance. These mechanisms can help manage log drift and reduce false positive signals. Feedback mechanisms are implemented in two primary ways: through iterative model improvements (e.g., network weight updates) or corrective filtering mechanisms external to the model. Filtering mechanisms include rule-based overrides on model outputs and input filtering at the preprocessor level. Our systematic literature review of online parser-supported log anomaly detection methods revealed only two studies that included feedback mechanisms (Table 6). Both were implemented as model update functions [17], [117].

B. CATEGORIZATION OF METHODS

To verify our proposed taxonomy, we used it to classify the log anomaly detection methods discovered through our systematic literature review. Table 6 contains the results of this classification. Figure 12 shows the associated parser classifications.

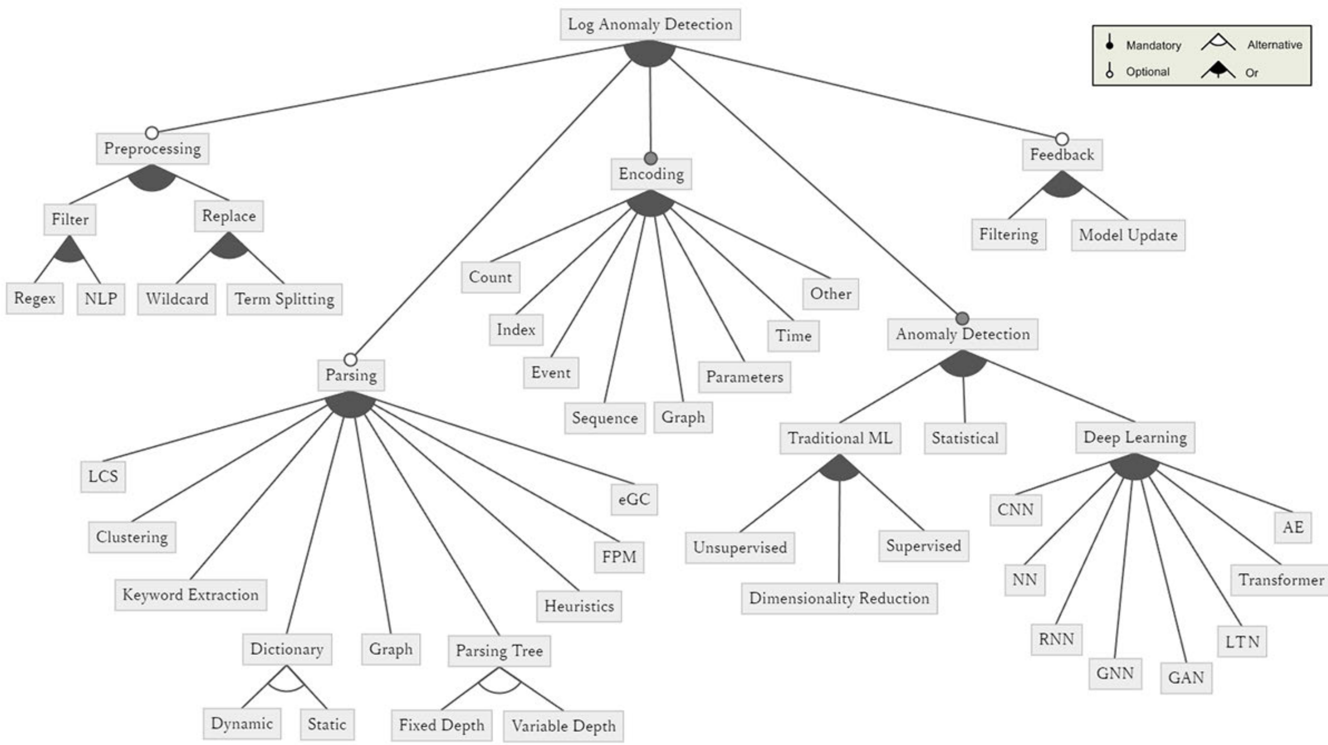

FIGURE 13. Online anomaly detection workflow taxonomy.

RQ4. How can different forms of online log anomaly detection be classified? **As online log anomaly detection workflows are composed of a combination of primary components, they can be easily classified by the existence and type of these components. Our proposed taxonomy provides a breakdown of currently available component types to create these categorizations (Fig. 13). Using this taxonomy, we successfully classified the online log anomaly detection studies discovered through our systematic literature review and verified the taxonomy's comprehensiveness. The resulting classifications are listed in Table 6.**

As can be seen from these results, Drain is by far the most commonly utilized parser. This trend has continued with modern studies since 2021 (Fig. 5). As for anomaly detection methods, deep learning approaches were the most frequently used (Fig. 3). 73% of these studies utilized sequence-based encoding, and sequences were produced using various techniques (Word2Vec, FastText, BERT, etc.). Preprocessors were used in 16% of the workflows, but it is worth noting that regex replacement style preprocessing was included in many others as part of their parser's data cleansing step. Only two studies implemented feedback mechanisms, both in the form of model update functions. This lack of feedback coverage is a significant research gap that merits future attention.

Through this classification, we confirmed that our proposed taxonomy is sufficient for categorizing the workflows available to date. However, the taxonomy will likely expand as research continues and new approaches are introduced. The addition of eGC as a parsing type (with the introduction of the eLP parser) is one example of this [197]. As research progresses, we will likely see more such expansions.

VIII. RESEARCH GAPS

While many studies introduce new methods, few explore component features and the intersection of their use against different forms of data. Research shortcomings discovered through our study include the lack of diversity in component utilization, the limited exploration of false positive mitigation strategies, the lack of real-world use case studies, and the sub-optimal assessment of log anomaly detection workflow robustness. These gaps are discussed in more detail in the following sections.

A. COMPONENT COMBINATIONS

Overall, Drain is the most frequently utilized online parsing method (Tables 1, 2, and 6). Although other online parsers have achieved higher performance (Fig. 8), they are used rarely in anomaly detection studies. Parser selection can significantly affect the performance of log anomaly detection workflows [168]. Han et al. demonstrated improvements in F-score values using Drain instead of Spell [89]. Fu et al. showed that while a "high parsing accuracy does not definitely imply high anomaly detection performance," anomaly detection methods performed more effectively and efficiently using heuristic-based log parsers [232]. Their

AQ:5

TABLE 6. Classification of online parser-supported log anomaly detection studies.

Study	Preprocessor	Parser	Encoding	Anomaly Detection Method	Feedback
AcLog, 2023 [106]	-	Drain	Sequence	LSTM (RNN)	-
Adaboost, 2022 [116]	-	Spell	Sequence (TF-IDF)	Adaboost (Supervised)	-
Adanomaly, 2022 [109]	-	Drain	Sequence	BiGAN (GAN)	-
ADR, 2020 [238]	-	Drain	Count	ADR (Statistical)	-
AdvGraLog, 2023 [97]	-	Drain	Graph	GAN	-
AllContext, 2020 [229]	Term Splitting	Drain	Sequence (ARE [241])	Bi-LSTM (RNN)	-
AllInfoLog, 2023 [91]	-	Drain	Sequence (RoBERTa) + Parameters/Time	Bi-LSTM (RNN)	-
ATT-GRU, 2020 [240]	-	Spell	Sequence or Parameters/Time	GRU (RNN)	-
BERT-Log, 2022 [74]	-	Drain	Sequence (BERT)	NN	-
BLATR, 2023 [89]	-	Drain	Sequence (TF-IDF)	Bi-LSTM (RNN)	-
CANet, 2022 [114]	-	FT-tree	Event (Word2Vec/NER)	NN	-
Cheansunan *et al.*, 2019 [241]	-	Drain	Index	CNN	-
Chen *et al.*, 2021 [66]	-	Drain	Count + Sequence (TF-IDF/Glove)	CNN/LSTM (RNN)	-
CL2MLog, 2022 [64]	-	Drain	Sequence (BERT)	Transformer	-
CLog, 2022 [55]	-	Drain	Sequence (NN+Att)	HMM (Statistical)	-
De La Torre Parra *et al.*, 2022 [123]	-	Spell	Sequence	Transformer	-
DeepLog, 2017 [17]	-	Spell	Sequence + Parameters/Time	LSTM (RNN)	Model Update
DeepLog + DCG, 2019 [242]	-	Spell	Index	LSTM (RNN)	-
ESAD, 2023 [100]	-	Drain	Sequence	LSTM (RNN) + k-means (Supervised)	-
ETCNLog, 2023 [92]	-	Drain	Sequence (Word2Vec)	TCN (CNN)	-
Fält *et al.*, 2022 [102]	-	Drain	Sequence (Word2Vec)	Transformer/LSTM (RNN)	-
FLOGCNN, 2021 [50]	-	Drain	Sequence	CNN	-
GLAD-PAW, 2021 [54]	-	Drain	Graph	GNN	-
Gulmez *et al.*, 2023 [122]	-	Spell	Sequence	CNN	-
HilBERT, 2023 [94]	Term Splitting§	Drain	Sequence (WordPiece)	BERT (Transformer)	-
Himler *et al.*, 2023 [120]	-	Spell	Sequence	LSTM	-
HitAnomaly, 2020 [243]	-	Drain	Sequence (BERT) + Event/Parameters	Transformer	-
Huy-Trung *et al.*, 2023 [60]	-	Drain	Index + Parameters + Time	RF/kNN/EGB (Supervised), NN/AE	-
IELog, 2023 [57]	-	Drain	Count + Sequence (GloVe/TF-IDF)	Ensemble (GNN/CNN/RNN/Supervised)	-
InterpretableSAD, 2021 [76]	-	Drain	Sequence (Word2Vec)	LSTM (RNN)	-
LADDERS, 2023 [58]	-	Drain	Sequence (TF-IDF)	Ensemble (Supervised/Unsupervised)	-
LCC-HGLog, 2022 [71]	NLP Filter + Term Splitting‡	Drain	Graph	GNN	-
Li, 2023 [113]	-	FT-tree	Count (IDF)	Clustering (Unsupervised)	-
Li & Su, 2023 [112]	-	FT-tree	Count (IDF)	Clustering (Unsupervised)	-
LogAttention, 2021 [51]	-	Drain	Sequence (FastText/TF-IDF)	NN [+Att]	-
LogBASA, 2023 [96]	-	Drain	Sequence (BERT) + Graph + Time	Transformer	-
LogBERT, 2021 [48]	-	Drain	Index	Transformer	-
LogBD, 2023 [95]	-	Drain	Sequence (BERT)	TCN (CNN)	-
LogBP-LORA, 2023 [93]	-	Drain	Sequence	BERT (Transformer)	-
LogC, 2020 [236]	-	Drain	Index + Other (Component Label)	LSTM (RNN)	-
LogCAD, 2022 [59]	-	Drain	Count	Adaboost/RF (Supervised)	-
LogEncoder, 2023 [86]	-	Drain	Sequence (BERT)	LSTM (RNN)	-
LogFlash, 2021 [46]	-	Drain	Sequence	TCFG (Statistical)	-
LogGD, 2022 [103]	-	Drain	Graph	GNN	-
LogKT, 2023 [83]	-	Drain	Sequence (WordPiece)	Transformer + Bi-LSTM (RNN)	-
LogLR, 2022 [72]	NLP Filter + Term Splitting‡	Drain	Sequence (FastText/TF-IDF/LSTM)	LTN	-
LogLS, 2022 [117]	-	Spell	Index	LSTM (RNN)	Model Update
LogNL, 2020 [244]	NLP Filter	Drain	Sequence (TF-IDF) + Parameters/Time	LSTM (RNN)	-
LogOnline, 2023 [107]	Term Splitting‡	Drain	Sequence (FastText) + Time + Other	AE + LSTM (RNN)	-
LogPal, 2023 [115]	NLP Filter†	FT-tree	Sequence (GloVe)	Transformer	-
LogPS, 2022 [82]	-	Drain	Sequence (Word2Vec/PoS)	Bi-LSTM (RNN)	-
LogRobust, 2019 [226]	NLP Filter + Term Splitting‡	Drain	Sequence (FastText/TF-IDF)	Bi-LSTM (RNN)	-
LSADNET, 2021 [69]	-	Various[11]	Sequence (Word2Vec/CNN)	Transformer	-
LTAnomaly, 2023 [80]	-	Drain	Sequence (Word2Vec/TF-IDF) + Parameters	Transformer	-
Macková *et al.*, 2023 [98]	-	Drain/Spell	Sequence	Transformer/CNN/LSTM (RNN)	-
MLog, 2023 [90]	-	Drain	Count + Sequence (BERT/IDF)	CNN + LSTM (RNN)	-
OC4Seq, 2021 [73]	-	Various[12]	Index	GRU (RNN)	-
OLSTM, 2023 [87]	-	Drain	Event (BERT)	LSTM (RNN)	-
OpenLog, 2022 [108]	Term Splitting‡	Drain	Sequence (GloVe)	CNN + Bi-LSTM (RNN)	-
PLELog, 2021 [101]	Term Splitting‡	Drain	Sequence (Word2Vec/TF-IDF)	GRU (RNN)	-
Puranik *et al.*, 2023 [104]	-	Drain	Sequence	Ensemble (RNN/CNN)	-
RADT, 2022 [78]	-	Drain	Sequence (FastText/TF-IDF)	Transformer	-
Ryciak *et al.*, 2022 [111]	RegEx Filter	Drain3	Sequence (FastText)	NN	-
S3M, 2023 [61]	-	Drain	Count + Sequence (+ Parameters)	AE	-
sADR/uADR, 2022 [44]	-	Drain	Count	ADR (Statistical)	-
SigML, 2023 [56]	-	Drain	Event	LR/SVM (Supervised)	-
SigML++, 2023 [84]	-	Drain	Event	LR/SVM (Supervised) [+NN]	-
Sinha *et al.*, 2022 [63]	-	Drain	Sequence	CNN	-
Sutthipanyo *et al.*, 2023 [99]	-	Drain	Sequence (Word2Vec)	CNN	-
SwissLog, 2020 [23]	Term Splitting§	SwissLog	Sequence (BERT) + Time	Bi-LSTM (RNN)	-
Tan *et al.*, 2023 [85]	-	Drain	Sequence (FastText/TF-IDF)	LSTM/GRU (RNN)	-
TCN-Log2Vec, 2023 [124]	-	TCN-Log2Vec	Sequence (BERT)	TCN (CNN)	-
Ts-CNN-Lin, 2022 [77]	-	Drain	Sequence + Time	CNN	-
Xiao *et al.*, 2022 [245]	-	Spell	Sequence (CNN)	LSTM (RNN)	-
Yan *et al.*, 2023 [119]	-	Spell	Graph	GNN + GAN	-
Yang *et al.*, 2023 [79]	-	Drain	Count + Sequence	Bi-GRU (RNN)	-
Yang *et al.*, 2023 [81]	-	Drain	Count + Sequence	Bi-GRU (RNN)	-
Yu *et al.*, 2021 [75]	NLP Filter	Drain	Count + Sequence (FastText/TF-IDF)	Bi-LSTM (RNN)	-
Zaojian *et al.*, 2023 [121]	-	Spell	Sequence (BERT)	Bi-LSTM (RNN)	-
Zeufack *et al.*, 2022 [227]	-	Drain	Count	OPTICS (Unsupervised)	-
Zhang *et al.*, 2022 [43]	-	Drain	Sequence (Bi-LSTM)	PCA (Dimensionality Reduction)	-
Zhang *et al.*, 2023 [105]	-	Drain	Sequence (Word2Vec/PoS)	CNN + LSTM (RNN)	-

‡ Camel Case [248] § Word Ninja † Torchtext

[11] Selects the best parser by performance against each dataset (amongst the Spell, Drain, and FT-Tree online parsers).
[12] Uses Drain for the BGL dataset and Spell for the HDFS dataset.

findings suggest that some combinations of parsers and anomaly detection methods lead to more optimal outcomes, but a limited number of parsers were considered. Le and Zhang also found that "the performance of models is highly influenced by log parsers" [247]. Some combinations handled noise better than others. For example, parsers such as Drain, which tend to overproduce log events, can hinder forecast-style, event sequence prediction approaches to log anomaly detection.

Combinations of other component types may also result in different performances. Xingfang et al. found that differing log representations have "a non-negligible influence" on downstream model effectiveness, but there exists "no single log representation technique that performs the best across all models and datasets" [7]. Similarly, combinations of different preprocessors, filters, and feedback mechanisms may also introduce different advantages. The evaluation of the intersection of these components merits future exploration for this reason.

In Table 6, we present a categorization of online parser-supported log anomaly detection methods discovered through our systematic literature reviews. This categorization was performed using our newly introduced taxonomy from Section VII. It reveals some interesting findings. First, methods with the highest F-scores (within the top ten) for both the HDFS and BGL datasets all use deep learning. They also use either semantic sequencing or graphical feature encoding. A mixture of preprocessing components and log parsers are used, but Drain is the most frequently applied parsing method overall. This evaluation, however, is still incomplete in terms of component coverage. A more comprehensive analysis of different component combinations against public and industrial datasets would be beneficial. Such a study could reveal insights into the strengths and weaknesses of component combinations and help guide achievable improvements to the overall accuracy of log anomaly detection pipelines.

B. FEEDBACK MECHANISMS

Feedback mechanisms provide a return route for corrective adjustments to log anomaly detection models. They allow for incremental improvements to model accuracy and reductions in false positives. They are also a key approach for managing log drift. However, our recent work revealed that the effectiveness of mitigating drift via current feedback methods with sequence-based anomaly detection models is limited [248]. These findings suggest that more extensive research on these topics is needed.

Du et al. introduced an unlearning framework that uses "a new objective function that aims to maximize the loss to unlearn reported abnormal samples" [249]. DeepLog uses an incremental process to update LSTM weights using corrected false positive signals provided by domain experts [17]. The DeepLog study found that simply increasing the amount of training data from one to ten percent did not significantly increase model precision. However, predictions and F-score values improved when incremental feedback updates were applied, regardless of the amount of data used in the initial training phase. These findings show that feedback mechanisms could be even more important than training data quantity for increasing model accuracy. Overall, however, very few anomaly detection studies have incorporated such mechanisms (Table 6).

Feedback mechanisms have the potential to improve the effectiveness of log anomaly detection workflows significantly. They are also a key approach for managing the degradation of model quality post-deployment. The lack of coverage of these mechanisms can be considered a significant research gap, and work to fill this void is an important area for future focus.

C. REAL-WORLD USE CASE STUDIES

Another significant log anomaly detection research gap is the lack of real-world use case studies. Log anomaly detection methods have been assessed mainly with a select number of public datasets. However, Petrescu et al. reveal that "industry logs are typically heterogeneous, thus threatening the applicability of log parsing in practice" [15]. There have been several log anomaly detection implementations used in industrial settings and research initiatives. Antić et al. introduced LOMOS, a solution functioning "in the context of supply chain resilience" that seeks to discover anomalous behavior that rule-based solutions may miss (implemented as an extension of LogBERT using the Drain parser) [250]. DeCorus-NSA is a solution developed by IBM for data center syslog monitoring [110]. There remains, however, a severe lack of evaluative studies on log anomaly detection methods in industrial settings.

Currently, rule-based monitoring approaches dominate the industry. These methods do not scale well against varied log sources, and their use can be burdensome [251]. However, like rule-based methods, log anomaly detection approaches also have strengths and weaknesses. In our previous work, we compared rule-based methods to anomaly detection methods using an industry dataset [228]. We found that while anomaly detection methods were more accurate, rule-based methods proved superior in practicality. The rule-based method was capable of detecting the evaluated incident with minimal delay and without producing false positives. While this evaluation was performed offline with only a single incident type, it reveals the need to better assess anomaly detection methods using real-world data.

Log anomaly detection literature commonly focuses on the accuracy and robustness of methods. However, more practical factors such as setup time, maintenance effort, running costs, and explainability are poorly studied. Real-world use case studies can help bridge these gaps and provide a more holistic picture of the challenges and benefits associated with applying log anomaly detection methods to real-world problems.

D. MODEL ROBUSTNESS MEASURES

Many online log parsers have been assessed for robustness using the 16 public log datasets in LogPAI's Loghub. Some parsers, however, have not been evaluated to this extent. As mentioned in Section V-B, the HDFS and BGL datasets are the most commonly used to assess parser performances. However, these datasets (HDFS in particular) have relatively few unique templates [16]. Preferably, parsers should be evaluated against a more diverse collection of log entries (including real-world industry data) and assessed with better metrics. Such metrics should include, for example, the stricter form of PA that considers the proper identification of dynamic parameters.

Log anomaly detection method studies suffer from these issues even more as most evaluations have only used a small number of publicly available datasets. As with parsers, assessing these methods using average performance measures across a diverse collection of data would be informative and useful. It would better reveal the methods' ability to deal with differing data sources, prove their real-world usability, and help reveal areas for further development.

RQ5. Does existing online parser-supported log anomaly detection research contain gaps that merit future exploration? **Gaps in log anomaly detection research include the lack of thorough component combination evaluations, exploration of feedback mechanisms, real-world use case studies, and robustness assessments. Addressing these gaps could have a significant impact on the real-world usability of log anomaly detection methods. For this reason, they deserve future focus and attention.**

IX. RELATED WORK

This section introduces an overview of peripheral topics related to online parser-supported log anomaly detection. These topics are beyond this study's scope but are significant research areas adjacent to our work. All studies presented were discovered through our systematic literature review described in Section III-A.

A. FEDERATED LEARNING

The majority of log anomaly detection studies discovered through our review covered single-process solutions. However, some work also explored federated and parallel approaches. De La Torre Parra et al. introduced a method of generating global federated learning models through the aggregation of local transformer-based model parameters [123]. Similarly, Shin and Kim introduced a federated learning framework that uses a global server to average and update aggregated weights from local site deep-learning models [161]. Guo et al. introduced a lightweight federated learning approach called FLOGCNN, attempting to address distributed log anomaly detection concerns such as bandwidth and privacy issues [50]. Wittkopp and Acker introduced a decentralized, federated learning method to synchronize distributed models trained on local data using model student and teacher roles [252].

Yang et al. introduced a distributed processing method for large-scale logs using Spark Streaming [81]. With this approach, they were able to improve the efficiency of parsing the HDFS dataset with Drain. Henriques et al. evaluated performance improvements using Dask [135]. They found that parallel processing outperformed their sequential approach to log anomaly detection even when using only two workers on a single node with two cores.

B. TRANSFER LEARNING

Some studies have explored log anomaly detection model transfer learning. These studies aim to develop workflows that can detect anomalies from multiple systems and mitigate cold-start issues when targeting new log sources. Chen et al. explored these topics with the introduction of LogTransfer, a framework that utilizes fully connected networks for anomaly classification between source and target systems [253]. Han and Yuan proposed an alternative approach called LogTAD. Their method performs transferable log anomaly detection without requiring labeled anomaly records from both the source and target systems [65].

LogTAD draws inspiration from the Deep Support Vector Data Description (Deep SVDD) method. Deep SVDD is a form of deep one-class classification that aims to model ''normality'' by ''minimizing the volume of a hypersphere that encloses the network representations of the data'' [254]. Huang et al. introduced a method for transfer learning using pseudo labels, annotations, and model training on unlabeled target data using a source classifier [255]. Finally, Liu et al. introduced LogBD, a method that uses domain adaptation to apply knowledge learned from source systems to target systems, "enabling the detection model to detect anomalies from multiple systems." [95].

C. HYPERPARAMETER TUNING

Log parsers and anomaly detection models generally require the tuning of hyperparameters to maximize their performance. This parameter tuning is often performed manually or through grid search. Improvements can be realized, however, through the use of algorithmic tuning. Marlaithong et al. proposed one such method, using the Artificial Bee Colony (ABC) algorithm to optimize the three key hyperparameters of the Drain parser [256]. Zhang et al. introduced the use of Population Based Training (PBT) to optimize PoS weight coefficients and anomaly detection model hyperparameters through parallel model training [105]. These methods can reduce the effort needed to configure log anomaly workflow parameters. They can also contribute to improvements in overall model performance.

D. SURVEYS

Zhaoxue et al. conducted a literature review of ''log processing in the context of AIOps and big data'' [257]. They examined log enhancement, parsing, and analysis. Although they included a summary of a selection of offline/online log parsers and log anomaly detection methods, they did not compare specific accuracy metrics. Zhang et al. performed a general survey on log parsing, providing a performance comparison of 17 open-source solutions (five of them being online methods) against the 16 public log datasets available in LogPAI's Loghub [18]. They presented a categorization of parsing methods consisting of four core types: clustering, frequent pattern mining, heuristics, and program analysis.

He et al. reviewed automated log analysis research, including sections covering several log parser and anomaly detection model characteristics [258]. They addressed log feature extraction types but did not perform accuracy comparisons. Ma et al. reviewed system log features utilized for log analysis and touched upon parsing methodology [235]. They presented comparative accuracy scores for several online and offline log parsers. Their categorization method for log features is utilized within our own log anomaly detection method taxonomy presented in Section VII.

Zhao, Jiang, and Ma introduced a basic framework for log anomaly detection and summarized recent detection models and technologies [234]. Their survey categorized encoding types, anomaly detection methods, and a selection of online and offline log parsers. Landauer et al. performed a systematic literature review of deep learning for anomaly detection in log data [19]. They reviewed deep learning-based log anomaly detection studies, summarizing the algorithmic approaches and encoding details of the workflows covered. Their review omits parser associations and performance statistics. Le and Zhang performed numerous experiments using deep learning log anomaly detection methods to analyze the impact of training data strategies, grouping approaches, class distributions, and data noise [247]. They found that these factors can significantly impact anomaly detection performance and provided observations on the nature of this impact.

X. THREATS TO VALIDITY

In this section, potential threats to validity are considered. Subsection X-A discusses internal threats, while X-B covers external ones.

A. THREATS TO INTERNAL VALIDITY

This survey compiles online log parser and log anomaly detection method evaluation results from multiple studies. Because workflow configurations can subtly differ between evaluations, the outcomes may vary. Although care was taken to minimize these differences using identical public datasets and metrics, factors such as hyperparameter usage and the log entry distribution amongst training and test sets could affect the results (and consequently, our own comparison of these reported results). Whenever possible, cross-checking of evaluations from multiple studies was performed, and only commonly reported outcomes from these experiments were utilized.

B. THREATS TO EXTERNAL VALIDITY

The performance of online log parsers and online parser-supported log anomaly detection methods can differ significantly depending on the targeted dataset. For online parsers, this threat has been mitigated to some extent by evaluating parser performance against a diverse collection of public log data (i.e., the 16 log datasets available in LogPAI's Loghub). There have been few assessments, however, using proprietary, industry-specific logs. For this reason, the performance achievable against these log targets may differ significantly from what has been presented.

Log anomaly detection methods may also be susceptible to variations in performance. Performance differences against other forms of log data for these methods are more likely given the lack of evaluative studies on their robustness. To date, most log anomaly detection studies have utilized only a select number of public log datasets for evaluation. Additionally, the log datasets often differ between studies. For this reason, it may be difficult to generalize the reported results to new data forms.

Finally, the studies covered in this survey focus primarily on log anomaly detection targets such as system errors, irregular states, and exceptions. The performance achieved when applying these methods to other anomaly detection targets may differ. Differences in performance may also be observed when applying these methods to unstructured data outside of the log domain.

XI. CONCLUSION AND FUTURE WORK

Log anomaly detection workflows heavily utilize online parsers. Of those used, Drain remains the most popular even though recent parsers have been shown to achieve higher average PA (Table 6 and Fig. 8). Drain is open source and performs significantly better than previous methods. This is likely one key reason for its continued popularity, even with higher-performing methods now available.

Of the studies surveyed since 2021, 84% used deep learning techniques, highlighting a shift from traditional machine learning approaches (Fig. 3). As the F-score values for anomaly detection methods have generally improved over time, there may be some data-based justification. However, ensemble methods using weak classifiers have performed as well as or better than deep learning methods in some evaluations (e.g., Adaboost using the HDFS dataset, as illustrated in Figure 9) [116]. These results suggest that while deep learning methods do show significant potential, there is merit in exploring traditional machine learning algorithms in future experiments as well. Exploring traditional machine learning approaches is not just for the performance potential but also to avoid the inherent weaknesses that plague deep learning methods (such as heavy resource utilization and long training times).

While log parsing remains heavily utilized in log anomaly detection research, direct vectorization approaches for log anomaly detection are becoming more prevalent (being used in 21% of the studies discovered since 2021 as shown in

Fig. 7). The popularity of online log parsing, however, significantly contributes to the highly accurate results produced by modern log anomaly detection workflows. Since 2021, 46 new parsing methods have been introduced (Table 3). 37% of these methods are online implementations (Fig. 6). The average PA of online parsers has steadily increased over time (Fig. 8). In 2023, Brain achieved the highest average PA recorded for the 16 log datasets in LogPAI's Loghub (0.981), improving upon SwissLog's score of 0.962 achieved in 2020 [23], [192].

Anomaly detection approaches have also shown gradual improvements in performance. These improvements are apparent when comparing results from individual evaluations against a common set of public log datasets. Since the introduction of DeepLog in 2017, the F-score values of new methods using the HDFS dataset have steadily increased (Fig. 9). Although these improvements have been small and gradual, they are noteworthy given the high performance achieved by DeepLog originally. Similar performance comparisons against the BGL dataset show historically high levels of achieved F-score values (Fig. 10). As research of new methods has continued, these advancements in performance have as well.

Online parser-supported log anomaly detection methods are built from a collection of fundamental components, but they differ significantly in their type and arrangement. Our taxonomy categorizes methods based on these components (Fig. 13). Using the taxonomy, we classified all anomaly detection workflows discovered through our systematic literature review (Table 6). This categorization shows common trends in research and can be used to guide experimentation with more diversified component sets in the future.

Log anomaly detection research has some gaps, including the lack of comparative studies on different combinations of workflow components, limited exploration of feedback mechanisms, the lack of real-world use case studies, and insufficient anomaly detection method robustness assessments. Research efforts to fill these gaps through future work would be beneficial. The main directions for this work should include in-depth comparative studies on combinations of anomaly detection workflow components and datasets (with better robustness measures), real-world use case assessments of these workflows, and the development and evaluation of false-positive mitigation strategies.

Parser selection may significantly impact the accuracy of log anomaly detection models [69]. Hence, a comprehensive, comparative study combining different online parsers with high-performing anomaly detection techniques would be useful. An evaluation of anomaly detection model robustness using a large, shared set of diverse log data would also be advantageous. This data should include public datasets and real-world industry log data containing real-world incidents. The robustness of models is a critical factor for industrial use, and there is a need to properly measure and account for it.

Such an experiment could be performed using our component-based online anomaly detection pipeline framework [228]. Representative log anomaly detection component types (as defined through our taxonomy) could be implemented as new *Encoder* and *Decoder* classes, and evaluated in all possible combinations against an extended collection of public and private log data. For parser performance, the stricter form of PA (i.e., requiring all dynamic parameters to be identified for a template to be considered correctly parsed) should be used to better represent parsing quality. The results of such an experiment would be extremely informative. These extensions to the framework would also prove useful for future researchers.

Online parser-supported log anomaly detection methods eliminate the need for manual rule setup and maintenance and have the potential to better detect unforeseen issues. Significant improvements in system estate reliability could be realized through the use of these methods and through the continued advancement of the technologies that support them. Our goal in performing this study is to encourage and promote such improvements through the analysis of the current state of these technologies and the provision of direction for future research.

ACKNOWLEDGMENT

The authors would like to thank Xingfang Wu from Polytechnique Montréal and the members of the Data Science Initiative (DSI), Nomura Securities Company Ltd., for their support in preparing and completing this research. This work was supported in part by the Japan Society for the Promotion of Science (JSPS) KAKENHI Program under Grant 21KK0179 and Grant 23K18470, and in part by the Japan Science and Technology Agency (JST)-Mirai Program under Grant JPMJMI20B8.

REFERENCES

[1] K. Zhang, J. Xu, M. R. Min, G. Jiang, K. Pelechrinis, and H. Zhang, "Automated IT system failure prediction: A deep learning approach," in *Proc. IEEE Int. Conf. Big Data (Big Data)*, Dec. 2016, pp. 1291–1300.

[2] S. Saleem, M. Sheeraz, M. Hanif, and U. Farooq, "Web server attack detection using machine learning," in *Proc. Int. Conf. Cyber Warfare Secur. (ICCWS)*, Oct. 2020, pp. 1–7.

[3] N. Shahid and M. Ali Shah, "Anomaly detection in system logs in the sphere of digital economy," in *Proc. Competitive Advantage Digit. Economy (CADE)*, Jun. 2021, pp. 185–190.

[4] U. Ünal and H. Dag, "AnomalyAdapters: Parameter-efficient multi-anomaly task detection," *IEEE Access*, vol. 10, pp. 5635–5646, 2022.

[5] D. Gunter, B. L. Tierney, A. Brown, M. Swany, J. Bresnahan, and J. M. Schopf, "Log summarization and anomaly detection for troubleshooting distributed systems," in *Proc. 8th IEEE/ACM Int. Conf. Grid Comput.*, Sep. 2007, pp. 226–234.

[6] J. Shi, G. He, and X. Liu, "Anomaly detection for key performance indicators through machine learning," in *Proc. Int. Conf. Netw. Infrastruct. Digit. Content (IC-NIDC)*, Aug. 2018, pp. 1–5.

[7] X. Wu, H. Li, and F. Khomh, "On the effectiveness of log representation for log-based anomaly detection," *Empirical Softw. Eng.*, vol. 28, no. 6, p. 137, Oct. 2023, doi: 10.1007/s10664-023-10364-1.

[8] W. Xu, L. Huang, A. Fox, D. Patterson, and M. I. Jordan, "Detecting large-scale system problems by mining console logs," in *Proc. ACM SIGOPS 22nd Symp. Operating Syst. Princ.*, Oct. 2009, pp. 37–44.

[9] A. A. O. Makanju, A. N. Zincir-Heywood, and E. E. Milios, "Clustering event logs using iterative partitioning," in *Proc. 15th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Jun. 2009, pp. 1255–1263.

[10] Q. Fu, J.-G. Lou, Y. Wang, and J. Li, "Execution anomaly detection in distributed systems through unstructured log analysis," in *Proc. 9th IEEE Int. Conf. Data Mining*, Dec. 2009, pp. 149–158.

[11] S. Messaoudi, A. Panichella, D. Bianculli, L. Briand, and R. Sasnauskas, "A search-based approach for accurate identification of log message formats," in *Proc. IEEE/ACM 26th Int. Conf. Program Comprehension (ICPC)*, May 2018, pp. 167–16710.

[12] M. Mizutani, "Incremental mining of system log format," in *Proc. IEEE Int. Conf. Services Comput.*, Jun. 2013, pp. 595–602.

[13] M. Du and F. Li, "Spell: Online streaming parsing of large unstructured system logs," *IEEE Trans. Knowl. Data Eng.*, vol. 31, no. 11, pp. 2213–2227, Nov. 2019.

[14] P. He, J. Zhu, Z. Zheng, and M. R. Lyu, "Drain: An online log parsing approach with fixed depth tree," in *Proc. IEEE Int. Conf. Web Services (ICWS)*, Jun. 2017, pp. 33–40.

[15] S. Petrescu, F. Den Hengst, A. Uta, and J. S. Rellermeyer, "Log parsing evaluation in the era of modern software systems," in *Proc. IEEE 34th Int. Symp. Softw. Rel. Eng. (ISSRE)*, Oct. 2023, pp. 379–390.

[16] J. Zhu, S. He, J. Liu, P. He, Q. Xie, Z. Zheng, and M. R. Lyu, "Tools and benchmarks for automated log parsing," in *Proc. IEEE/ACM 41st Int. Conf. Softw. Eng., Softw. Eng. Pract. (ICSE-SEIP)*, May 2019, pp. 121–130.

[17] M. Du, F. Li, G. Zheng, and V. Srikumar, "DeepLog: Anomaly detection and diagnosis from system logs through deep learning," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur.* New York, NY, USA: Association for Computing Machinery, Oct. 2017, pp. 1285–1298.

[18] T. Zhang, H. Qiu, G. Castellano, M. Rifai, C. S. Chen, and F. Pianese, "System log parsing: A survey," *IEEE Trans. Knowl. Data Eng.*, vol. 35, no. 8, pp. 8596–8614, Jan. 2023.

[19] M. Landauer, S. Onder, F. Skopik, and M. Wurzenberger, "Deep learning for anomaly detection in log data: A survey," *Mach. Learn. With Appl.*, vol. 12, Jun. 2023, Art. no. 100470. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2666827023000233

[20] S. Lupton, H. Washizaki, N. Yoshioka, and Y. Fukazawa, "Online log parsing: Preliminary literature review," in *Proc. IEEE Int. Symp. Softw. Rel. Eng. Workshops (ISSREW)*, Wuhan, China, Oct. 2021, pp. 304–305.

[21] S. Lupton, H. Washizaki, N. Yoshioka, and Y. Fukazawa, "Literature review on log anomaly detection approaches utilizing online parsing methodology," in *Proc. 28th Asia–Pacific Softw. Eng. Conf. (APSEC)*, Dec. 2021, pp. 559–563.

[22] M. Du and F. Li, "Spell: Streaming parsing of system event logs," in *Proc. IEEE 16th Int. Conf. Data Mining (ICDM)*, Dec. 2016, pp. 859–864.

[23] X. Li, P. Chen, L. Jing, Z. He, and G. Yu, "SwissLog: Robust and unified deep learning based log anomaly detection for diverse faults," in *Proc. IEEE 31st Int. Symp. Softw. Rel. Eng. (ISSRE)*, Oct. 2020, pp. 92–103.

[24] X. Li, P. Chen, L. Jing, Z. He, and G. Yu, "SwissLog: Robust anomaly detection and localization for interleaved unstructured logs," *IEEE Trans. Depend. Secure Comput.*, vol. 20, no. 4, pp. 2762–2780, Aug. 2023.

[25] S. Zhang, W. Meng, J. Bu, S. Yang, Y. Liu, D. Pei, J. Xu, Y. Chen, H. Dong, X. Qu, and L. Song, "Syslog processing for switch failure diagnosis and prediction in datacenter networks," in *Proc. IEEE/ACM 25th Int. Symp. Quality Service (IWQoS)*, Jun. 2017, pp. 1–10.

[26] H. Dai, H. Li, C.-S. Chen, W. Shang, and T.-H. Chen, "Logram: Efficient log parsing using *n*n-gram dictionaries," *IEEE Trans. Softw. Eng.*, vol. 48, no. 3, pp. 879–892, Mar. 2022.

[27] T. Kimura, A. Watanabe, T. Toyono, and K. Ishibashi, "Proactive failure detection learning generation patterns of large-scale network logs," in *Proc. 11th Int. Conf. Netw. Service Manage. (CNSM)*, Nov. 2015, pp. 8–14.

[28] T. Kimura, A. Watanabe, T. Toyono, and K. Ishibashi, "Proactive failure detection learning generation patterns of large-scale network logs," *IEICE Trans. Commun.*, vol. E102.B, no. 2, pp. 306–316, 2019.

[29] S. Huang, Y. Liu, C. Fung, R. He, Y. Zhao, H. Yang, and Z. Luan, "Paddy: An event log parsing approach using dynamic dictionary," in *Proc. NOMS IEEE/IFIP Netw. Oper. Manage. Symp.*, Apr. 2020, pp. 1–8.

[30] S. Zhang, Y. Liu, W. Meng, J. Bu, S. Yang, Y. Sun, D. Pei, J. Xu, Y. Zhang, L. Song, and M. Zhang, "Efficient and robust syslog parsing for network devices in datacenter networks," *IEEE Access*, vol. 8, pp. 30245–30261, 2020.

[31] A. Agrawal, R. Karlupia, and R. Gupta, "Logan: A distributed online log parser," in *Proc. IEEE 35th Int. Conf. Data Eng. (ICDE)*, Apr. 2019, pp. 1946–1951.

[32] R. Yang, D. Qu, Y. Qian, Y. Dai, and S. Zhu, "An online log template extraction method based on hierarchical clustering," *EURASIP J. Wireless Commun. Netw.*, vol. 2019, no. 1, pp. 1–12, Dec. 2019.

[33] X. Duan, S. Ying, H. Cheng, W. Yuan, and X. Yin, "OILog: An online incremental log keyword extraction approach based on MDP-LSTM neural network," *Inf. Syst.*, vol. 95, Jan. 2021, Art. no. 101618.

[34] Y. Zhong, Y. Guo, and C. Liu, "FLP: A feature-based method for log parsing," *Electron. Lett.*, vol. 54, no. 23, pp. 1334–1336, Nov. 2018.

[35] X. Wang, Y. Zhao, H. Xiao, X. Wang, and X. Chi, "LTmatch: A method to abstract pattern from unstructured log," *Appl. Sci.*, vol. 11, no. 11, p. 5302, Jun. 2021.

[36] Z. Chunyong and X. Meng, "Log parser with one-to-one markup," in *Proc. 3rd Int. Conf. Inf. Comput. Technol. (ICICT)*, Mar. 2020, pp. 251–257.

[37] S. Guo, Z. Liu, W. Chen, and T. Li, "Event extraction from streaming system logs," in *Proc. Inf. Sci. Appl. (ICISA)*, vol. 514, 2019, pp. 465–474.

[38] P. Wen, Z. Zhang, and B. Deng, "OLMPT: Research on online log parsing method based on prefix tree," in *Proc. 3rd Int. Conf. Inf. Technol. Electr. Eng.*, Hunan, China, Dec. 2020, pp. 55–59.

[39] Z. Zhao, C. Wang, and W. Rao, "Slop: Towards an efficient and universal streaming log parser," in *Information and Communications Security*, D. Naccache, S. Xu, S. Qing, P. Samarati, G. Blanc, R. Lu, Z. Zhang, and A. Meddahi, Eds. Cham, Switzerland: Springer, 2018, pp. 325–341.

[40] K. Shima, "Length matters: Clustering system log messages using length of words," 2016, *arXiv:1611.03213*.

[41] D. A. Bhanage, A. V. Pawar, and K. Kotecha, "IT infrastructure anomaly detection and failure handling: A systematic literature review focusing on datasets, log preprocessing, machine & deep learning approaches and automated tool," *IEEE Access*, vol. 9, pp. 156392–156421, 2021.

[42] Z. Jian, Z. Jin, X. Xie, Y. Lu, G. Li, X. Chen, and T. Baker, "Sysnif: A log-based workflow construction method and performance measurement in intelligent IoT system," *Measurement*, vol. 186, Dec. 2021, Art. no. 110175.

[43] Y. Zhang, D. Zhang, F. Guo, X. Wang, Y. Duan, and X. Zhang, "Log anomaly detection based on bi-LSTM feature extraction," in *Proc. 5th Int. Conf. Data Sci. Inf. Technol. (DSIT)*, Jul. 2022, pp. 1–6.

[44] B. Zhang, H. Zhang, V.-H. Le, P. Moscato, and A. Zhang, "Semi-supervised and unsupervised anomaly detection by mining numerical workflow relations from system logs," *Automated Softw. Eng.*, vol. 30, no. 1, p. 4, May 2023.

[45] J. Nyyssölä, M. Mäntylä, and M. Varela, "How to configure masked event anomaly detection on software logs?" in *Proc. IEEE Int. Conf. Softw. Maintenance Evol. (ICSME)*, Oct. 2022, pp. 414–418.

[46] T. Jia, Y. Wu, C. Hou, and Y. Li, "LogFlash: Real-time streaming anomaly detection and diagnosis from system logs for large-scale software systems," in *Proc. IEEE 32nd Int. Symp. Softw. Rel. Eng. (ISSRE)*, Oct. 2021, pp. 80–90.

[47] M. Mäntylä, M. Varela, and S. Hashemi, "Pinpointing anomaly events in logs from stability testing—N-Grams vs. deep-learning," in *Proc. IEEE Int. Conf. Softw. Test., Verification Validation Workshops (ICSTW)*, Apr. 2022, pp. 285–292.

[48] H. Guo, S. Yuan, and X. Wu, "LogBERT: Log anomaly detection via BERT," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, Jul. 2021, pp. 1–8. [Online]. Available: https://api.semanticscholar.org/CorpusID:232146842

[49] H. Studiawan and F. Sohel, "Anomaly detection in a forensic timeline with deep autoencoders," *J. Inf. Secur. Appl.*, vol. 63, Dec. 2021, Art. no. 103002.

[50] Y. Guo, Y. Wu, Y. Zhu, B. Yang, and C. Han, "Anomaly detection using distributed log data: A lightweight federated learning approach," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, Jul. 2021, pp. 1–8.

[51] Q. Du, L. Zhao, J. Xu, Y. Han, and S. Zhang, "Log-based anomaly detection with multi-head scaled dot-product attention mechanism," in *Proc. Database Expert Syst. Appl., 32nd Int. Conf.*, vol. 12923, 2021, pp. 335–347.

[52] N. Zhao, J. Chen, Z. Yu, H. Wang, J. Li, B. Qiu, H. Xu, W. Zhang, K. Sui, and D. Pei, "Identifying bad software changes via multimodal anomaly detection for online service systems," in *Proc. 29th ACM Joint Meeting Eur. Softw. Eng. Conf. Symp. Found. Softw. Eng.* New York, NY, USA: Association for Computing Machinery, Aug. 2021, pp. 527–539, doi: 10.1145/3468264.3468543.

[53] H. Ott, J. Bogatinovski, A. Acker, S. Nedelkoski, and O. Kao, "Robust and transferable anomaly detection in log data using pre-trained language models," in *Proc. IEEE/ACM Int. Workshop Cloud Intell. (CloudIntelligence)*. Los Alamitos, CA, USA: IEEE Computer Society, May 2021, pp. 19–24.

[54] Y. Wan, Y. Liu, D. Wang, and Y. Wen, "GLAD-PAW: Graph-based log anomaly detection by position aware weighted graph attention network," in *Proc. Pacific–Asia Conf. Knowl. Discovery Data Mining*, vol. 12712, 2021, pp. 66–77.

[55] J. Bogatinovski, S. Nedelkoski, L. Wu, J. Cardoso, and O. Kao, "Failure identification from unstable log data using deep learning," in *Proc. 22nd IEEE Int. Symp. Cluster, Cloud Internet Comput. (CCGrid)*, May 2022, pp. 346–355.

[56] D. Trivedi, A. Boudguiga, and N. Triandopoulos, "SigML: Supervised log anomaly with fully homomorphic encryption," in *Proc. Int. Symp. Cyber Secur., Cryptol., Mach. Learn.*, vol. 13914. Cham, Switzerland: Springer, 2023, pp. 372–388, doi: 10.1007/978-3-031-34671-2_26.

[57] W. Xiong, W. Chen, J. Liu, and K. Zhao, "An anomaly detection framework for system logs based on ensemble learning," in *PRICAI 2023: Trends in Artificial Intelligence*, F. Liu, A. A. Sadanandan, D. N. Pham, P. Mursanto, and D. Lukose, Eds. Singapore: Springer, 2024, pp. 52–65.

[58] S. A. Mondal, P. Rv, S. Rao, and A. Menon, "LADDERS: Log based anomaly detection and diagnosis for enterprise systems," *Ann. Data Sci.*, vol. 10, pp. 1–19, Jun. 2023. [Online]. Available: https://link.springer.com/article/10.1007/s40745-023-00471-7#citeas and https://link.springer.com/journal/40745/volumes-and-issues

[59] C. Liu, M. Liang, J. Hou, Z. Gu, and Z. Wang, "LogCAD: An efficient and robust model for log-based conformal anomaly detection," *Secur. Commun. Netw.*, vol. 2022, pp. 1–13, Mar. 2022.

[60] N. Huy-Trung and N. Viet Quoc, "Anomaly detection in Internet of Things based on logs using machine learning and deep learning techniques," in *Proc. 4th Int. Conf. Comput., Netw. Internet Things*. New York, NY, USA: Association for Computing Machinery, May 2023, pp. 969–974, doi: 10.1145/3603781.3604223.

[61] G. Horváth, A. Kádár, and P. Szilágyi, "The sub-sequence summary method for detecting anomalies in logs," *IEEE Access*, vol. 11, pp. 37412–37423, 2023.

[62] J. Bogatinovski and S. Nedelkoski, "Multi-source anomaly detection in distributed it systems," in *Proc. Int. Conf. Service-Oriented Comput.*, vol. 12632, 2021, pp. 201–213.

[63] R. Sinha, R. Sur, R. Sharma, and A. K. Shrivastava, "Anomaly detection using system logs: A deep learning approach," *Int. J. Inf. Secur. Privacy*, vol. 16, no. 1, pp. 1–15, Nov. 2021.

[64] X. Wang, Q. Cao, Q. Wang, Z. Cao, X. Zhang, and P. Wang, "Robust log anomaly detection based on contrastive learning and multi-scale MASS," *J. Supercomput.*, vol. 78, no. 16, pp. 17491–17512, Nov. 2022.

[65] X. Han and S. Yuan, "Unsupervised cross-system log anomaly detection via domain adaptation," in *Proc. 30th ACM Int. Conf. Inf. Knowl. Manage.* New York, NY, USA: Association for Computing Machinery, Oct. 2021, pp. 3068–3072, doi: 10.1145/3459637.3482209.

[66] H. Chen, "Unsupervised anomaly detection based on system logs," in *Proc. 33rd Int. Conf. Softw. Eng. Knowl. Eng.*, Jul. 2021, pp. 92–97, doi: 10.18293/seke2021-126.

[67] Y. Wang and Z. Ji, "Design and implementation of a semi-supervised anomaly log detection model DDA," in *Proc. Int. Conf. Comput. Commun. Artif. Intell. (CCAI)*, May 2021, pp. 86–90.

[68] N. Zhao, H. Wang, Z. Li, X. Peng, G. Wang, Z. Pan, Y. Wu, Z. Feng, X. Wen, W. Zhang, K. Sui, and D. Pei, "An empirical investigation of practical log anomaly detection for online service systems," in *Proc. 29th ACM Joint Meeting Eur. Softw. Eng. Conf. Symp. Found. Softw. Eng.* New York, NY, USA: Association for Computing Machinery, Aug. 2021, pp. 1404–1415, doi: 10.1145/3468264.3473933.

[69] C. Zhang, X. Wang, H. Zhang, H. Zhang, and P. Han, "Log sequence anomaly detection based on local information extraction and globally sparse transformer model," *IEEE Trans. Netw. Service Manage.*, vol. 18, no. 4, pp. 4119–4133, Dec. 2021.

[70] M. Zhang, J. Chen, J. Liu, J. Wang, R. Shi, and H. Sheng, "LogST: Log semi-supervised anomaly detection based on sentence-BERT," in *Proc. 7th Int. Conf. Signal Image Process. (ICSIP)*, Jul. 2022, pp. 356–361.

[71] Y. Fang, Z. Zhao, Y. Xu, and Z. Liu, "Log anomaly detection based on hierarchical graph neural network and label contrastive coding," *Comput., Mater. Continua*, vol. 74, no. 2, pp. 4099–4118, 2023.

[72] K. Zhang, X. Di, X. Liu, B. Li, L. Fang, Y. Qin, and J. Cao, *LogLR: A Log Anomaly Detection Method Based on Logical Reasoning* (Lecture Notes in Computer Science), vol. 13472. Cham, Switzerland: Springer, 2022.

[73] Z. Wang, Z. Chen, J. Ni, H. Liu, H. Chen, and J. Tang, "Multi-scale one-class recurrent neural networks for discrete event sequence anomaly detection," in *Proc. 27th ACM SIGKDD Conf. Knowl. Discovery Data Mining*, Singapore, Aug. 2021, pp. 3726–3734.

[74] S. Chen and H. Liao, "BERT-log: Anomaly detection for system logs based on pre-trained language model," *Appl. Artif. Intell.*, vol. 36, no. 1, Dec. 2022, Art. no. 2145642.

[75] D. Yu, X. Hou, C. Li, Q. Lv, Y. Wang, and N. Li, "Anomaly detection in unstructured logs using attention-based bi-LSTM network," in *Proc. 7th IEEE Int. Conf. Netw. Intell. Digit. Content (IC-NIDC)*, Nov. 2021, pp. 403–407.

[76] X. Han, H. Cheng, D. Xu, and S. Yuan, "InterpretableSAD: Interpretable anomaly detection in sequential log data," in *Proc. IEEE Int. Conf. Big Data (Big Data)*, Dec. 2021, pp. 1183–1192.

[77] Y. Huangfu, S. Habibi, and A. Wassyng, "System failure detection using deep learning models integrating timestamps with nonuniform intervals," *IEEE Access*, vol. 10, pp. 17629–17640, 2022.

[78] Y. Chen, L. Ye, Y. Ye, P. Zhang, and Q. Tan, "Anomaly detection from log data sequences with perturbations," in *Proc. 7th IEEE Int. Conf. Data Sci. Cyberspace (DSC)*, Jul. 2022, pp. 183–190.

[79] H. Yang, F. Lin, Y. Chai, K. Qie, W. Lin, Y. Wang, C. Zhang, and M. Guo, "An anomaly detection algorithm for logs based on self-attention mechanism and bigru model," in *Proc. Chin. Intell. Syst. Conf.*, Y. Jia, W. Zhang, Y. Fu, and J. Wang, Eds. Singapore: Springer, 2023, pp. 877–888.

[80] D. Han, M. Sun, M. Li, and Q. Chen, "LTAnomaly: A transformer variant for syslog anomaly detection based on multi-scale representation and long sequence capture," *Appl. Sci.*, vol. 13, no. 13, p. 7668, Jun. 2023. [Online]. Available: https://www.mdpi.com/2076-3417/13/13/7668

[81] H. Yang, F. Lin, Y. Chai, K. Qie, S. Li, Y. Wang, C. Zhang, and M. Guo, "A distributed processing method for large scale logs," in *Proc. Chin. Intell. Syst. Conf.*, Oct. 2023, pp. 515–524.

[82] D. Li, J. Zhang, X. Zhang, F. Lin, C. Wang, and L. Chang, "LogPS: A robust log sequential anomaly detection approach based on natural language processing," in *Proc. IEEE 22nd Int. Conf. Commun. Technol. (ICCT)*, Nov. 2022, pp. 1400–1405.

[83] X. Ou and J. Liu, "LogKT: Hybrid log anomaly detection method for cloud data center," in *Proc. IEEE 47th Annu. Comput., Softw., Appl. Conf. (COMPSAC)*, Jun. 2023, pp. 164–173.

[84] D. Trivedi, A. Boudguiga, N. Kaaniche, and N. Triandopoulos, "SigML++: Supervised log anomaly with probabilistic polynomial approximation," *Cryptography*, vol. 7, no. 4, p. 52, Oct. 2023. [Online]. Available: https://www.mdpi.com/2410-387X/7/4/52

[85] Y. Tan, J. Wang, J. Liu, and Y. Li, "Deep learning-based log anomaly detection for 5G core network," in *Proc. IEEE/CIC Int. Conf. Commun. China (ICCC)*, Dalian, China, Aug. 2023, pp. 1–6, doi: 10.1109/iccc57788.2023.10233555.

[86] 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 Trans. Netw. Service Manage.*, vol. 20, no. 2, pp. 1378–1391, Jan. 2023.

[87] D. A. Bhanage, A. V. Pawar, K. Kotecha, and A. Abraham, "Failure detection using semantic analysis and attention-based classifier model for IT infrastructure log data," *IEEE Access*, vol. 11, pp. 108178–108197, 2023.

[88] S. Tao, Y. Liu, W. Meng, Z. Ren, H. Yang, X. Chen, L. Zhang, Y. Xie, C. Su, X. Oiao, W. Tian, Y. Zhu, T. Han, Y. Qin, and Y. Li, "Biglog: Unsupervised large-scale pre-training for a unified log representation," in *Proc. IEEE/ACM 31st Int. Symp. Quality Service (IWQoS)*, Jun. 2023, pp. 1–11.

[89] P. Han, H. Li, G. Xue, and C. Zhang, "Distributed system anomaly detection using deep learning-based log analysis," *Comput. Intell.*, vol. 39, no. 3, pp. 433–455, Jun. 2023, doi: 10.1111/coin.12573.

[90] Y. Fu, K. Liang, and J. Xu, "MLog: Mogrifier LSTM-based log anomaly detection approach using semantic representation," *IEEE Trans. Services Comput.*, vol. 16, no. 5, pp. 3537–3549, Jun. 2023.

[91] R. Xiao, H. Chen, J. Lu, W. Li, and S. Jin, "AllInfoLog: Robust diverse anomalies detection based on all log features," *IEEE Trans. Netw. Service Manage.*, vol. 20, no. 3, pp. 2529–2543, Jun. 2023.

[92] Y. Chang, N. Luktarhan, J. Liu, and Q. Chen, "ETCNLog: A system log anomaly detection method based on efficient channel attention and temporal convolutional network," *Electronics*, vol. 12, no. 8, p. 1877, Apr. 2023. [Online]. Available: https://www.mdpi.com/2079-9292/12/8/1877

[93] S. He, Y. Lei, Y. Zhang, K. Xie, and P. K. Sharma, "Parameter-efficient log anomaly detection based on pre-training model and LoRa," in *Proc. IEEE 34th Int. Symp. Softw. Rel. Eng. (ISSRE)*, Oct. 2023, pp. 207–217.

[94] S. Huang, Y. Liu, C. Fung, H. Wang, H. Yang, and Z. Luan, "Improving log-based anomaly detection by pre-training hierarchical transformers," *IEEE Trans. Comput.*, vol. 72, no. 9, pp. 2656–2667, Mar. 2023.

[95] S. Liu, L. Deng, H. Xu, and W. Wang, "LogBD: A log anomaly detection method based on pretrained models and domain adaptation," *Appl. Sci.*, vol. 13, no. 13, p. 7739, Jun. 2023. [Online]. Available: https://www.mdpi.com/2076-3417/13/13/7739

[96] L. Liao, K. Zhu, J. Luo, and J. Cai, "LogBASA: Log anomaly detection based on system behavior analysis and global semantic awareness," *Int. J. Intell. Syst.*, vol. 2023, pp. 1–18, Sep. 2023, doi: 10.1155/2023/3777826.

[97] Z. He, W. Chen, Y. Tang, K. Zhao, and J. Liu, "Graph-based log anomaly detection via adversarial training," in *Proc. Int. Symp. Dependable Softw. Eng., Theories, Tools, Appl.*, Dec. 2023, pp. 55–71.

[98] K. Macková, D. Benk, and M. Šrotýř, "Comparative analysis of deep learning models and preprocessing techniques for anomaly detection in syslog," in *Proc. 14th Int. Conf. Inf. Commun. Syst. (ICICS)*, Nov. 2023, pp. 1–6.

[99] T. Sutthipanyo, T. Lamsan, W. Thawornsusin, and W. Susutti, "Log-based anomaly detection using CNN model with parameter entity labeling for improving log preprocessing approach," in *Proc. TENCON IEEE Region Conf. (TENCON)*, Oct. 2023, pp. 914–919.

[100] H. Cheng, D. Xu, and S. Yuan, "Explainable sequential anomaly detection via prototypes," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, Jun. 2023, pp. 1–8.

[101] 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 *Proc. IEEE/ACM 43rd Int. Conf. Softw. Eng. (ICSE)*, May 2021, pp. 1448–1460.

[102] M. Fält, S. Forsström, Q. He, and T. Zhang, "Learning-based anomaly detection using log files with sequential relationships," in *Proc. 6th Int. Conf. Syst. Rel. Saf. (ICSRS)*, Nov. 2022, pp. 337–342.

[103] Y. Xie, H. Zhang, and M. A. Babar, "LogGD: Detecting anomalies from system logs with graph neural networks," in *Proc. IEEE 22nd Int. Conf. Softw. Quality, Rel. Secur. (QRS)*, Dec. 2022, pp. 299–310.

[104] A. Puranik, A. V. Akkihal, R. K. Suhas, Y. P. D. Patel, and H. B. Mahesh, "Ensemble deep learning based real-time log anomaly detection," in *Proc. Int. Conf. Digit. Appl., Transformation Economy (ICDATE)*, Jul. 2023, pp. 1–7.

[105] X. Zhang, J. Zhang, J. Yang, F. Lin, C. Wang, L. Chang, and D. Li, "An anomaly detection approach of part-of-speech log sequence via population based training," in *Proc. IEEE 3rd Int. Conf. Power, Electron. Comput. Appl. (ICPECA)*, Jan. 2023, pp. 254–258.

[106] C. Duan, T. Jia, Y. Li, and G. Huang, "AcLog: An approach to detecting anomalies from system logs with active learning," in *Proc. IEEE Int. Conf. Web Services (ICWS)*, Jul. 2023, pp. 436–443.

[107] X. Wang, J. Song, X. Zhang, J. Tang, W. Gao, and Q. Lin, "LogOnline: A semi-supervised log-based anomaly detector aided with online learning mechanism," in *Proc. 38th IEEE/ACM Int. Conf. Automated Softw. Eng. (ASE)*, Sep. 2023, pp. 141–152.

[108] Z. Xu, Z. Jiang, T. Li, J. You, B. Wu, and L. Li, "OpenLog: Incremental anomaly classification with changing, unbalanced and unknown logs," in *Proc. IEEE Int. Conf. Parallel Distrib. Process. With Appl., Big Data Cloud Comput., Sustain. Comput. Commun., Social Comput. Netw. (ISPA/BDCloud/SocialCom/SustainCom)*, Dec. 2022, pp. 571–580.

[109] J. Qi, Z. Luan, S. Huang, Y. Wang, C. Fung, H. Yang, and D. Qian, "Adanomaly: Adaptive anomaly detection for system logs with adversarial learning," in *Proc. NOMS IEEE/IFIP Netw. Oper. Manage. Symp.*, Apr. 2022, pp. 1–5.

[110] D. Ohana, B. Wassermann, M. Hershcovitch, E. K. Kolodner, M. Malka, E. Raichstein, R. Schaffer, and R. Shahla, "DeCorus-NSA: Detection and correlation of unusual signals for network syslog analytics," in *Proc. 14th ACM Int. Conf. Syst. Storage*. New York, NY, USA: Association for Computing Machinery, Jun. 2021, p. 1, doi: 10.1145/3456727.3463827.

[111] P. Ryciak, K. Wasielewska, and A. Janicki, "Anomaly detection in log files using selected natural language processing methods," *Appl. Sci.*, vol. 12, no. 10, p. 5089, May 2022.

[112] Y. Li and Y. Su, "The insider threat detection method of university website clusters based on machine learning," in *Proc. 6th Int. Conf. Artif. Intell. Big Data (ICAIBD)*, May 2023, pp. 560–565.

[113] Y. Li, "Improved insider threat detection method of university cluster system based on log-clustering," in *Proc. 6th Int. Conf. Big Data Technol.* New York, NY, USA: Association for Computing Machinery, Sep. 2023, pp. 236–241, doi: 10.1145/3627377.3627414.

[114] Z. Li, J. Zhang, X. Zhang, F. Lin, C. Wang, and X. Cai, "Natural language processing-based model for log anomaly detection," in *Proc. IEEE 2nd Int. Conf. Softw. Eng. Artif. Intell. (SEAI)*, Jun. 2022, pp. 129–134.

[115] L. Sun and X. Xu, "LogPal: A generic anomaly detection scheme of heterogeneous logs for network systems," *Secur. Commun. Netw.*, vol. 2023, pp. 1–12, Apr. 2023, doi: 10.1155/2023/2803139.

[116] B. Wang, Q. Hua, H. Zhang, X. Tan, Y. Nan, R. Chen, and X. Shu, "Research on anomaly detection and real-time reliability evaluation with the log of cloud platform," *Alexandria Eng. J.*, vol. 61, no. 9, pp. 7183–7193, Sep. 2022.

[117] Y. Chen, N. Luktarhan, and D. Lv, "LogLS: Research on system log anomaly detection method based on dual LSTM," *Symmetry*, vol. 14, no. 3, p. 454, Feb. 2022.

[118] S. Gu, Y. Chu, W. Zhang, P. Liu, Q. Yin, and Q. Li, "Research on system log anomaly detection combining two-way slice GRU and GA-attention mechanism," in *Proc. 4th Int. Conf. Artif. Intell. Big Data (ICAIBD)*, May 2021, pp. 577–583.

[119] L. Yan, C. Luo, and R. Shao, "Discrete log anomaly detection: A novel time-aware graph-based link prediction approach," *Inf. Sci.*, vol. 647, Nov. 2023, Art. no. 119576. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0020025523011611

[120] P. Himler, M. Landauer, F. Skopik, and M. Wurzenberger, "Towards detecting anomalies in log-event sequences with deep learning: Open research challenges," in *Proc. Eur. Interdiscipl. Cybersecurity Conf.* New York, NY, USA: Association for Computing Machinery, Jun. 2023, pp. 71–77, doi: 10.1145/3590777.3590789.

[121] D. Zaojian, L. Yong, C. Mu, C. Liang, F. Wengao, and L. Ziang, "Semi-supervised power microservices log anomaly detection based on BiLSTM and BERT with attention," in *Proc. 2nd Int. Conf. Adv. Electron., Electr. Green Energy (AEEGE)*, May 2023, pp. 82–87.

[122] S. Gulmez, A. Duman, S. A. Duman, and I. Sogukpinar, "Log mining-based online failure prediction in client-server architecture," in *Proc. 8th Int. Conf. Comput. Sci. Eng. (UBMK)*, Sep. 2023, pp. 492–497.

[123] G. De La Torre Parra, L. Selvera, J. Khoury, H. Irizarry, E. Bou-Harb, and P. Rad, "Interpretable federated transformer log learning for cloud threat forensics," in *Proc. Netw. Distrib. Syst. Secur. Symp.*, 2022, pp. 1–16. [Online]. Available: https://api.semanticscholar.org/CorpusID:248221052

[124] Z. Jiang, Y. Gao, J. Yuan, K. Yuan, and X. Li, "TCN-Log2 Vec: A comprehensive log anomaly detection framework based on optimized log parsing and temporal convolutional network," in *Proc. 3rd Int. Conf. Consum. Electron. Comput. Eng. (ICCECE)*, Jan. 2023, pp. 515–519.

[125] K. Ahmed, A. Altaf, N. S. M. Jamail, F. Iqbal, and R. Latif, "ADAL-NN: Anomaly detection and localization using deep relational learning in distributed systems," *Appl. Sci.*, vol. 13, no. 12, p. 7297, Jun. 2023. [Online]. Available: https://www.mdpi.com/2076-3417/13/12/7297

[126] M. Li, M. Sun, G. Li, D. Han, and M. Zhou, "MDFULog: Multi-feature deep fusion of unstable log anomaly detection model," *Appl. Sci.*, vol. 13, no. 4, p. 2237, Feb. 2023. [Online]. Available: https://www.mdpi.com/2076-3417/13/4/2237

[127] X. Liu, W. Liu, X. Di, J. Li, B. Cai, W. Ren, and H. Yang, "LogNADS: Network anomaly detection scheme based on log semantics representation," *Future Gener. Comput. Syst.*, vol. 124, pp. 390–405, Nov. 2021.

[128] X. Duan, S. Ying, W. Yuan, H. Cheng, and X. Yin, "A generative adversarial networks for log anomaly detection," *Comput. Syst. Sci. Eng.*, vol. 37, no. 1, pp. 135–148, 2021.

[129] T. Xiao, Z. Quan, Z.-J. Wang, Y. Le, Y. Du, X. Liao, K. Li, and K. Li, "Loader: A log anomaly detector based on transformer," *IEEE Trans. Services Comput.*, vol. 16, no. 5, pp. 3479–3492, Sep. 2023.

[130] Y. Zhou and Y. Su, "Polo: Adaptive trie-based log parser for anomaly detection," *Mathematics*, vol. 11, no. 23, p. 4797, Nov. 2023. [Online]. Available: https://www.mdpi.com/2227-7390/11/23/4797

[131] K. A. Alharthi, A. Jhumka, S. Di, F. Cappello, and E. Chuah, "Sentiment analysis based error detection for large-scale systems," in *Proc. 51st Annu. IEEE/IFIP Int. Conf. Dependable Syst. Netw. (DSN)*, Jun. 2021, pp. 237–249.
[132] V. Dentamaro, V. N. Convertini, S. Galantucci, P. Giglio, T. Palmisano, and G. Pirlo, "Ensemble consensus: An unsupervised algorithm for anomaly detection in network security data," in *Proc. Italian Conf. Cybersecurity*, 2021, pp. 309–318. [Online]. Available: https://api.semanticscholar.org/CorpusID:244476710
[133] C. Cavallaro and E. Ronchieri, *Identifying Anomaly Detection Patterns From Log Files: A Dynamic Approach*, vol. 12950. Cham, Switzerland: Springer, 2021.
[134] A. Behera, C. R. Panigrahi, S. Behera, and B. Pati, *Anomaly Detection Unstructured Logs Generated From Complex Micro-Service Based Architecture Using One-Class SVM* (Lecture Notes in Networks and Systems), vol. 428. Singapore: Springer, 2023.
[135] J. Henriques, F. Caldeira, T. Cruz, and P. Simões, "Combining K-means and XGBoost models for anomaly detection using log datasets," *Electronics*, vol. 9, no. 7, p. 1164, Jul. 2020.
[136] L.-P. Yuan, P. Liu, and S. Zhu, "Recompose event sequences vs. predict next events: A novel anomaly detection approach for discrete event logs," in *Proc. ACM Asia Conf. Comput. Commun. Secur.* New York, NY, USA: Association for Computing Machinery, May 2021, pp. 336–348, doi: 10.1145/3433210.3453098.
[137] D. Zhang, D. Dai, R. Han, and M. Zheng, "SentiLog: Anomaly detecting on parallel file systems via log-based sentiment analysis," in *Proc. 13th ACM Workshop Hot Topics Storage File Syst.* New York, NY, USA: Association for Computing Machinery, Jul. 2021, pp. 86–93, doi: 10.1145/3465332.3470873.
[138] D. Zhu, H. Sun, N. Li, B. Mi, and T. Xi, "BS-net: A behavior sequence network for insider threat detection," in *Proc. IEEE Symp. Comput. Commun. (ISCC)*. Los Alamitos, CA, USA: IEEE Computer Society, Sep. 2021, pp. 1–6, doi: 10.1109/ISCC53001.2021.9631445.
[139] C. Egersdoerfer, D. Zhang, and D. Dai, "ClusterLog: Clustering logs for effective log-based anomaly detection," in *Proc. IEEE/ACM 12th Workshop Fault Tolerance HPC eXtreme Scale (FTXS)*, Nov. 2022, pp. 1–10.
[140] Q. Zhou, X. Dang, D. Huo, Q. Ruan, C. Li, Y. Wang, and Z. Xu, "LogBlock: An anomaly detection method on permissioned blockchain based on log-block sequence," in *Proc. IEEE Smartworld, Ubiquitous Intell. Comput., Scalable Comput. Commun., Digit. Twin, Privacy Comput., Metaverse, Auto. Trusted Vehicles (SmartWorld/UIC/ScalCom/DigitalTwin/PriComp/Meta)*, Dec. 2022, pp. 1008–1015.
[141] W. Wang, S. Lu, J. Luo, and C. Wu, "DeepUserLog: Deep anomaly detection on user log using semantic analysis and key-value data," in *Proc. IEEE 34th Int. Symp. Softw. Rel. Eng. (ISSRE)*, Oct. 2023, pp. 172–182.
[142] D. Zhang, C. Egersdoerfer, T. Mahmud, M. Zheng, and D. Dai, "Drill: Log-based anomaly detection for large-scale storage systems using source code analysis," in *Proc. IEEE Int. Parallel Distrib. Process. Symp. (IPDPS)*, May 2023, pp. 189–199.
[143] 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 *Proc. IEEE 34th Int. Symp. Softw. Rel. Eng. (ISSRE)*, Oct. 2023, pp. 355–366.
[144] S. Han, Q. Wu, H. Zhang, B. Qin, J. Hu, X. Shi, L. Liu, and X. Yin, "Log-based anomaly detection with robust feature extraction and online learning," *IEEE Trans. Inf. Forensics Security*, vol. 16, pp. 2300–2311, 2021.
[145] X. Duan, S. Ying, W. Yuan, H. Cheng, and X. Yin, "QLLog: A log anomaly detection method based on Q-learning algorithm," *Inf. Process. Manage.*, vol. 58, no. 3, May 2021, Art. no. 102540.
[146] M. Cinque, R. Della Corte, and A. Pecchia, "Micro2vec: Anomaly detection in microservices systems by mining numeric representations of computer logs," *J. Netw. Comput. Appl.*, vol. 208, Dec. 2022, Art. no. 103515.
[147] B. Tak, S. Park, and P. Kudva, "Priolog: Mining important logs via temporal analysis and prioritization," *Sustainability*, vol. 11, no. 22, p. 6306, Nov. 2019.
[148] M. Chnib and W. Gabsi, "Detection of anomalies in the HDFS dataset," in *Proc. IEEE/ACIS 21st Int. Conf. Softw. Eng. Res., Manage. Appl. (SERA)*, May 2023, pp. 243–250.
[149] B. Xia, Y. Bai, J. Yin, Y. Li, and J. Xu, "LogGAN: A log-level generative adversarial network for anomaly detection using permutation event modeling," *Inf. Syst. Frontiers*, vol. 23, no. 2, pp. 285–298, Apr. 2021.
[150] R. Hirakawa, H. Uchida, A. Nakano, K. Tominaga, and Y. Nakatoh, "Anomaly detection on software log based on temporal memory," *Comput. Electr. Eng.*, vol. 95, Oct. 2021, Art. no. 107433.
[151] Z. Wang, J. Tian, H. Fang, L. Chen, and J. Qin, "LightLog: A lightweight temporal convolutional network for log anomaly detection on the edge," *Comput. Netw.*, vol. 203, Feb. 2022, Art. no. 108616.
[152] J. Yu, Y. Jiao, Q. Guo, C. Liang, L. Rui, and X. Chen, "Design of log analysis system based on deep learning for operation system anomaly detection," in *Proc. 11th Int. Conf. Comput. Eng. Netw.*, vol. 808, 2022, pp. 884–892.
[153] M. Catillo, A. Pecchia, and U. Villano, "AutoLog: Anomaly detection by deep autoencoding of system logs," *Expert Syst. Appl.*, vol. 191, Apr. 2022, Art. no. 116263.
[154] Y. Xie, H. Zhang, B. Zhang, M. Babar, and S. Lu, "LogDP: Combining dependency and proximity for log-based anomaly detection," in *Proc. Int. Conf. Service-Oriented Comput.*, vol. 13121, pp. 708–716, 2021.
[155] R. Hirakawa, H. Uchida, A. Nakano, K. Tominaga, and Y. Nakatoh, "Large scale log anomaly detection via spatial pooling," *Cognit. Robot.*, vol. 1, pp. 188–196, Jan. 2021.
[156] T. Li, Y. Jiang, C. Lin, M. S. Obaidat, Y. Shen, and J. Ma, "DeepAG: Attack graph construction and threats prediction with bi-directional deep learning," *IEEE Trans. Depend. Secure Comput.*, vol. 20, no. 1, pp. 740–757, Jan. 2023.
[157] X. Wang, L. Yang, D. Li, L. Ma, Y. He, J. Xiao, J. Liu, and Y. Yang, "MADDC: Multi-scale anomaly detection, diagnosis and correction for discrete event logs," in *Proc. 38th Annu. Comput. Secur. Appl. Conf.*, Dec. 2022, pp. 769–784.
[158] W. Yunanto and H.-K. Pao, "User behaviour risk evaluation in zero trust architecture environment," in *Proc. IEEE 8th World Forum Internet Things (WF-IoT)*, Oct. 2022, pp. 1–6.
[159] C. Hu, X. Sun, H. Dai, H. Zhang, and H. Liu, "Research on log anomaly detection based on sentence-BERT," *Electronics*, vol. 12, no. 17, p. 3580, Aug. 2023. [Online]. Available: https://www.mdpi.com/2079-9292/12/17/3580
[160] T. Zhang, X. Huang, W. Zhao, S. Bian, and P. Du, "LogPrompt: A log-based anomaly detection framework using prompts," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, Jun. 2023, pp. 1–8.
[161] T.-H. Shin and S.-H. Kim, "Utility analysis about log data anomaly detection based on federated learning," *Appl. Sci.*, vol. 13, no. 7, p. 4495, Apr. 2023.
[162] W. Meng, Y. Liu, S. Zhang, F. Zaiter, Y. Zhang, Y. Huang, Z. Yu, Y. Zhang, L. Song, M. Zhang, and D. Pei, "LogClass: Anomalous log identification and classification with partial labels," *IEEE Trans. Netw. Service Manage.*, vol. 18, no. 2, pp. 1870–1884, Jun. 2021.
[163] J. Wang, C. Zhao, S. He, Y. Gu, O. Alfarraj, and A. Abugabah, "LogUAD: Log unsupervised anomaly detection based on Word2 Vec," *Comput. Syst. Sci. Eng.*, vol. 41, no. 3, pp. 1207–1222, 2022.
[164] A. Catovic, C. Cartwright, Y. T. Gebreyesus, and S. Ferlin, "Linnaeus: A highly reusable and adaptable ML based log classification pipeline," in *Proc. IEEE/ACM 1st Workshop AI Eng. Softw. Eng. AI (WAIN)*, May 2021, pp. 11–18.
[165] M. Hariharan, A. Mishra, S. Ravi, A. Sharma, A. Tanwar, K. Sundaresan, P. Ganesan, and R. Karthik, "Detecting log anomaly using subword attention encoder and probabilistic feature selection," *Int. J. Speech Technol.*, vol. 53, no. 19, pp. 22297–22312, Oct. 2023.
[166] Z. Qiu, Z. Zhou, B. Niblett, A. Johnston, J. Schwartzentruber, N. Zincir-Heywood, and M. I. Heywood, "Assessing the impact of bag-of-words versus word-to-vector embedding methods and dimension reduction on anomaly detection from log files," *Int. J. Netw. Manage.*, vol. 34, no. 1, Jan. 2024, Art. no. e2251, doi: 10.1002/nem.2251.
[167] Y. Wang and X. Li, "FastTransLog: A log-based anomaly detection method based on fastformer," in *Proc. 9th Int. Conf. Dependable Syst. Their Appl. (DSA)*, Aug. 2022, pp. 446–453.
[168] V.-H. Le and H. Zhang, "Log-based anomaly detection without log parsing," in *Proc. 36th IEEE/ACM Int. Conf. Automated Softw. Eng. (ASE)*, Nov. 2021, pp. 492–504.
[169] Z. Zhao, C. Xu, and B. Li, "A LSTM-based anomaly detection model for log analysis," *J. Signal Process. Syst.*, vol. 93, no. 7, pp. 745–751, Jul. 2021.

[170] L. Xi, Y. Xin, S. Luo, Y. Shang, and Q. Tang, "Anomaly detection mechanism based on hierarchical weights through large-scale log data," in *Proc. Int. Conf. Comput. Commun. Artif. Intell. (CCAI)*, May 2021, pp. 106–115.

[171] D. Lv, N. Luktarhan, and Y. Chen, "ConAnomaly: Content-based anomaly detection for system logs," *Sensors*, vol. 21, no. 18, p. 6125, Sep. 2021.

[172] L. Zhang, W. Li, Z. Zhang, Q. Lu, C. Hou, P. Hu, T. Gui, and S. Lu, "LogAttn: Unsupervised log anomaly detection with an autoencoder based attention mechanism," in *Proc. Logattn: Unsupervised Log Anomaly Detection With Autoencoder Based Attention Mechanism*, vol. 12817, 2021, pp. 222–235.

[173] C. Zhang, X. Wang, H. Zhang, J. Zhang, H. Zhang, C. Liu, and P. Han, "LayerLog: Log sequence anomaly detection based on hierarchical semantics," *Appl. Soft Comput.*, vol. 132, Jan. 2023, Art. no. 109860.

[174] K. A. Alharthi, A. Jhumka, S. Di, and F. Cappello, "Clairvoyant: A log-based transformer-decoder for failure prediction in large-scale systems," in *Proc. 36th ACM Int. Conf. Supercomputing*, Jun. 2022, pp. 1–14.

[175] F. Hang, W. Guo, H. Chen, L. Xie, C. Zhou, and Y. Liu, "Logformer: Cascaded transformer for system log anomaly detection," *Comput. Model. Eng. Sci.*, vol. 136, no. 1, pp. 517–529, 2023.

[176] M. Fält, S. Forsström, and T. Zhang, "Machine learning based anomaly detection of log files using ensemble learning and self-attention," in *Proc. 5th Int. Conf. Syst. Rel. Saf. (ICSRS)*, Nov. 2021, pp. 209–215.

[177] G. Tian, N. Luktarhan, H. Wu, and Z. Shi, "CLDTLog: System log anomaly detection method based on contrastive learning and dual objective tasks," *Sensors*, vol. 23, no. 11, p. 5042, May 2023. [Online]. Available: https://www.mdpi.com/1424-8220/23/11/5042

[178] X. Xie, S. Jian, C. Huang, F. Yu, and Y. Deng, "LogRep: Log-based anomaly detection by representing both semantic and numeric information in raw messages," in *Proc. IEEE 34th Int. Symp. Softw. Rel. Eng. (ISSRE)*, Oct. 2023, pp. 194–206.

[179] T. Wittkopp, D. Scheinert, P. Wiesner, A. Acker, and O. Kao, "PULL: Reactive log anomaly detection based on iterative PU learning," in *Proc. Annu. Hawaii Int. Conf. Syst. Sci.*, Jan. 2023, pp. 1376–1385. [Online]. Available: https://hdl.handle.net/10125/102802

[180] P. K. Mvula, P. Branco, G.-V. Jourdan, and H. L. Viktor, "HEART: Heterogeneous log anomaly detection using robust transformers," in *Discovery Science*, A. Bifet, A. C. Lorena, R. P. Ribeiro, J. Gama, and P. H. Abreu, Eds. Cham, Switzerland: Springer, 2023, pp. 673–687.

[181] Y. Lee, J. Kim, and P. Kang, "LAnoBERT: System log anomaly detection based on BERT masked language model," *Appl. Soft Comput.*, vol. 146, Oct. 2023, Art. no. 110689. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S156849462300707X

[182] R. Larisch, J. Vitay, and F. H. Hamker, "Detecting anomalies in system logs with a compact convolutional transformer," *IEEE Access*, vol. 11, pp. 113464–113479, 2023.

[183] G. Horváth, A. Mészáros, and P. Szilágyi, "TeleDAL: A regression-based template-less unsupervised method for finding anomalies in log sequences," *J. Supercomput.*, vol. 79, no. 16, pp. 18394–18416, Nov. 2023.

[184] S. Nam, E. Jeong, J. Hong, J.-H. Yoo, and J. W.-K. Hong, "Log analysis and prediction for anomaly detection in network switches," in *Proc. 19th Int. Conf. Netw. Service Manage. (CNSM)*, Oct. 2023, pp. 1–7.

[185] Y. Huo, C. Lee, Y. Su, S. Shan, J. Liu, and M. R. Lyu, "EvLog: Identifying anomalous logs over software evolution," in *Proc. IEEE 34th Int. Symp. Softw. Rel. Eng. (ISSRE)*, Oct. 2023, pp. 391–402.

[186] H. Guo, Y. Guo, J. Yang, J. Liu, Z. Li, T. Zheng, L. Zheng, W. Hou, and B. Zhang, *LogLG: Weakly Supervised Log Anomaly Detection via Log-Event Graph Construction*. Cham, Switzerland: Springer, Apr. 2023, pp. 490–501.

[187] T. Wittkopp, A. Acker, S. Nedelkoski, J. Bogatinovski, D. Scheinert, W. Fan, and O. Kao, "A2Log: Attentive augmented log anomaly detection," in *Proc. Annu. Hawaii Int. Conf. Syst. Sci.*, 2022, pp. 1–10.

[188] S. He, J. Zhu, P. He, and M. R. Lyu, "Loghub: A large collection of system log datasets towards automated log analytics," 2008, *arXiv:2008.06448*.

[189] H. Yang, D. Sun, Y. Wang, N. Zhao, S. Zhang, and W. Huang, "AdaptParse: Adaptive contextual aware attention network for log parsing via word classification," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, Jun. 2023, pp. 1–8.

[190] S. Jain, A. de Buitléir, and E. Fallon, "A framework for adaptive deep reinforcement semantic parsing of unstructured data," in *Proc. Int. Conf. Inf. Commun. Technol. Converg. (ICTC)*, Oct. 2021, pp. 1055–1060.

[191] V. Bertalan and D. Aloise, "Using transformer models and textual analysis for log parsing," in *Proc. IEEE 34th Int. Symp. Softw. Rel. Eng. (ISSRE)*. Los Alamitos, CA, USA: IEEE Computer Society, Oct. 2023, pp. 367–378, doi: 10.1109/issre59848.2023.00037.

[192] S. Yu, P. He, N. Chen, and Y. Wu, "Brain: Log parsing with bidirectional parallel tree," *IEEE Trans. Services Comput.*, vol. 16, no. 5, pp. 3224–3237, Jun. 2023.

[193] V.-H. Le and H. Zhang, "Log parsing: How far can ChatGPT go?" in *Proc. 38th IEEE/ACM Int. Conf. Automated Softw. Eng. (ASE)*, Sep. 2023, pp. 1699–1704.

[194] R. Tian, Z.-L. Diao, H.-Y. Jiang, and G.-G. Xie, "Cognition: Accurate and consistent linear log parsing using template correction," *J. Comput. Sci. Technol.*, vol. 38, no. 5, pp. 1036–1050, Sep. 2023, doi: 10.1007/s11390-021-1691-3.

[195] D. Plaisted and M. Xie, "DIP: A log parser based on 'disagreement index token' conditions," in *Proc. ACM Southeast Conf.*, Apr. 2022, pp. 113–122.

[196] Y. Fu, M. Yan, J. Xu, J. Li, Z. Liu, X. Zhang, and D. Yang, "Investigating and improving log parsing in practice," in *Proc. 30th ACM Joint Eur. Softw. Eng. Conf. Symp. Found. Softw. Eng.*, Nov. 2022, pp. 1566–1577.

[197] L. Decker, D. Leite, and D. Bonacorsi, "Explainable log parsing and online interval granular classification from streams of words," in *Proc. IEEE Int. Conf. Fuzzy Syst. (FUZZ-IEEE)*, Jul. 2022, pp. 1–8.

[198] A. Mandal, S. Gupta, S. Agarwal, and P. Mohapatra, "Improved topology extraction using discriminative parameter mining logs," in *Proc. Pacific–Asia Conf. Knowl. Discovery Data Mining*, May 2021, pp. 333–345.

[199] J. Xu, Q. Fu, Z. Zhu, Y. Cheng, Z. Li, Y. Ma, and P. He, "Hue: A user-adaptive parser for hybrid logs," in *Proc. 31st ACM Joint Eur. Softw. Eng. Conf. Symp. Found. Softw. Eng.* New York, NY, USA: Association for Computing Machinery, Nov. 2023, pp. 413–424, doi: 10.1145/3611643.3616260.

[200] M. Platini, T. Ropars, B. Pelletier, and N. De Palma, "LogFlow: Simplified log analysis for large scale systems," in *Proc. 22nd Int. Conf. Distrib. Comput. Netw.* New York, NY, USA: Association for Computing Machinery, Jan. 2021, pp. 116–125, doi: 10.1145/3427796.3427808.

[201] S. Yu, Y. Wu, Z. Li, P. He, N. Chen, and C. Liu, "Log parsing with generalization ability under new log types," in *Proc. 31st ACM Joint Eur. Softw. Eng. Conf. Symp. Found. Softw. Eng.* New York, NY, USA: Association for Computing Machinery, Nov. 2023, pp. 425–437, doi: 10.1145/3611643.3616355.

[202] T. Nguyen, S. Kobayashi, and K. Fukuda, "LogDTL: Network log template generation with deep transfer learning," in *Proc. IFIP/IEEE Int. Symp. Integr. Netw. Manage. (IM)*, May 2021, pp. 848–853.

[203] V.-H. Le and H. Zhang, "Log parsing with prompt-based few-shot learning," in *Proc. IEEE/ACM 45th Int. Conf. Softw. Eng. (ICSE)*, May 2023, pp. 2438–2449.

[204] S. Zhang and G. Wu, "Efficient online log parsing with log punctuations signature," *Appl. Sci.*, vol. 11, no. 24, p. 11974, Dec. 2021.

[205] P. Chen, G. Chao, L. Yang, H. He, L. Hong, M. Li, D. Gao, and S. Guo, "A robust log parsing algorithm—Practice of logslaw in heterogeneous logs of Pacific credit card center of bank of Communications(PCCC)," in *Proc. IEEE Int. Conf. Image Process. Comput. Appl. (ICIPCA)*, Aug. 2023, pp. 126–133.

[206] S. Tao, W. Meng, Y. Cheng, Y. Zhu, Y. Liu, C. Du, T. Han, Y. Zhao, X. Wang, and H. Yang, "LogStamp: Automatic online log parsing based on sequence labelling," *ACM SIGMETRICS Perform. Eval. Rev.*, vol. 49, no. 4, pp. 93–98, Jun. 2022.

[207] M. M. T and E. Baburaj, "Log file template detection as a multi-objective optimization problem," *Int. J. Swarm Intell. Res.*, vol. 13, no. 1, pp. 1–20, 2022.

[208] T. Marlaithong, V. C. Barroso, and P. Phunchongharn, "A log parsing framework for Alice o2 facilities," *IEEE Access*, vol. 11, pp. 69439–69457, 2023.

[209] Y.-Q. Zhu, J.-Y. Deng, J.-C. Pu, P. Wang, S. Liang, and W. Wang, "ML-parser: An efficient and accurate online log parser," *J. Comput. Sci. Technol.*, vol. 37, no. 6, pp. 1412–1426, Dec. 2022.

[210] Y. Bai, Y. Chi, and D. Zhao, "PatCluster: A top-down log parsing method based on frequent words," *IEEE Access*, vol. 11, pp. 8275–8282, 2023.

[211] M. Raynal, M.-O. Buob, and G. Quénot, "A novel pattern-based edit distance for automatic log parsing," in *Proc. 26th Int. Conf. Pattern Recognit. (ICPR)*, Aug. 2022, pp. 1236–1242.

[212] H. Dai, Y. Tang, H. Li, and W. Shang, "PILAR: Studying and mitigating the influence of configurations on log parsing," in *Proc. IEEE/ACM 45th Int. Conf. Softw. Eng. (ICSE)*, May 2023, pp. 818–829.

[213] G. Chu, J. Wang, Q. Qi, H. Sun, S. Tao, and J. Liao, "Prefix-graph: A versatile log parsing approach merging prefix tree with probabilistic graph," in *Proc. IEEE 37th Int. Conf. Data Eng. (ICDE)*, Apr. 2021, pp. 2411–2422.

[214] W. Yuan, S. Ying, X. Duan, H. Cheng, Y. Zhao, and J. Shang, "PVE: A log parsing method based on VAE using embedding vectors," *Inf. Process. Manage.*, vol. 60, no. 5, Sep. 2023, Art. no. 103476. https://www.sciencedirect.com/science/article/pii/S0306457323002133

[215] L. Fang, X. Di, X. Liu, Y. Qin, W. Ren, and Q. Ding, "QuickLogS: A quick log parsing algorithm based on template similarity," in *Proc. IEEE 20th Int. Conf. Trust, Secur. Privacy Comput. Commun. (TrustCom)*, Oct. 2021, pp. 1085–1092.

[216] S. Yu, N. Chen, Y. Wu, and W. Dou, "Self-supervised log parsing using semantic contribution difference," *J. Syst. Softw.*, vol. 200, Jun. 2023, Art. no. 111646.

[217] L. Sun and X. Xu, "SNNLog: A log parsing scheme with Siamese network and fixed depth tree in networks," in *Proc. 26th Int. Conf. Comput. Supported Cooperat. Work Design (CSCWD)*, May 2023, pp. 1025–1030.

[218] S. He, "SPINE: A scalable log parser with feedback guidance," in *Proc. 17th Innov. Softw. Eng. Conf.*, Feb. 2024, pp. 1198–1208.

[219] F. Zou, X. Chen, Y. Luo, T. Huang, Z. Liao, and K. Song, "Spray: Streaming log parser for real-time analysis," *Secur. Commun. Netw.*, vol. 2022, pp. 1–11, Sep. 2022.

[220] I. Sedki, A. Hamou-Lhadj, O. Ait-Mohamed, and M. A. Shehab, "An effective approach for parsing large log files," in *Proc. IEEE Int. Conf. Softw. Maintenance Evol. (ICSME)*, Oct. 2022, pp. 1–12.

[221] Y. Liu, X. Zhang, S. He, H. Zhang, L. Li, Y. Kang, Y. Xu, M. Ma, Q. Lin, Y. Dang, S. Rajmohan, and D. Zhang, "UniParser: A unified log parser for heterogeneous log data," in *Proc. ACM Web Conf.*, Apr. 2022, pp. 1893–1901.

[222] A. Vervaet, R. Chiky, and M. Callau-Zori, "USTEP: Unfixed search tree for efficient log parsing," in *Proc. IEEE Int. Conf. Data Mining (ICDM)*, Dec. 2021, pp. 659–668.

[223] A. Vervaet, M. Callau-Zori, Y. Chabchoub, and R. Chiky, "Online log parsing using evolving research tree," *Knowl. Inf. Syst.*, vol. 66, no. 2, pp. 1231–1255, Feb. 2024.

[224] Z. Li, C. Luo, T.-H. Chen, W. Shang, S. He, Q. Lin, and D. Zhang, "Did we miss something important? Studying and exploring variable-aware log abstraction," in *Proc. IEEE/ACM 45th Int. Conf. Softw. Eng. (ICSE)*, May 2023, pp. 830–842.

[225] S. Nedelkoski, J. Bogatinovski, A. Acker, J. Cardoso, and O. Kao, *Self-Supervised Log Parsing*. Cham, Switzerland: Springer, Feb. 2021, pp. 122–138.

[226] X. Zhang, Y. Xu, Q. Lin, B. Qiao, H. Zhang, Y. Dang, C. Xie, X. Yang, Q. Cheng, Z. Li, J. Chen, X. He, R. Yao, J.-G. Lou, M. Chintalapati, F. Shen, and D. Zhang, "Robust log-based anomaly detection on unstable log data," in *Proc. 27th ACM Joint Meeting Eur. Softw. Eng. Conf. Symp. Found. Softw. Eng.* New York, NY, USA: Association for Computing Machinery, Aug. 2019, pp. 807–817.

[227] V. Zeufack, D. Kim, D. Seo, and A. Lee, "An unsupervised anomaly detection framework for detecting anomalies in real time through network system's log files analysis," *High-Confidence Comput.*, vol. 1, no. 2, Dec. 2021, Art. no. 100030.

[228] S. Lupton, L. Yu, H. Washizaki, N. Yoshioka, and Y. Fukazawa, "Assessment of real-world incident detection through a component-based online log anomaly detection pipeline framework," in *Proc. 10th Int. Conf. Dependable Syst. Their Appl. (DSA)*, Tokyo, Japan, Aug. 2023, pp. 1–2.

[229] P. Sun, E. Yuepeng, T. Li, Y. Wu, J. Ge, J. You, and B. Wu, "Context-aware learning for anomaly detection with imbalanced log data," in *Proc. IEEE 22nd Int. Conf. High Perform. Comput. Commun., IEEE 18th Int. Conf. Smart City, IEEE 6th Int. Conf. Data Sci. Syst. (HPCC/SmartCity/DSS)*, Dec. 2020, pp. 449–456.

[230] A. Oliner and J. Stearley, "What supercomputers say: A study of five system logs," in *Proc. 37th Annu. IEEE/IFIP Int. Conf. Dependable Syst. Netw. (DSN)*, Jun. 2007, pp. 575–584.

[231] P. He, J. Zhu, S. He, J. Li, and M. R. Lyu, "An evaluation study on log parsing and its use in log mining," in *Proc. 46th Annu. IEEE/IFIP Int. Conf. Dependable Syst. Netw. (DSN)*, Jun. 2016, pp. 654–661.

[232] Y. Fu, M. Yan, Z. Xu, X. Xia, X. Zhang, and D. Yang, "An empirical study of the impact of log parsers on the performance of log-based anomaly detection," *Empirical Softw. Eng.*, vol. 28, no. 1, p. 6, Jan. 2023.

[233] R. Vaarandi, "A data clustering algorithm for mining patterns from event logs," in *Proc. 3rd IEEE Workshop IP Oper. Manage. (IPOM)*, Oct. 2003, pp. 119–126.

[234] X. Zhao, Z. Jiang, and J. Ma, "A survey of deep anomaly detection for system logs," in *Proc. Int. Joint Conf. Neural Netw.*, Jul. 2022, pp. 1–8.

[235] J. Ma, Y. Liu, H. Wan, and G. Sun, "Automatic parsing and utilization of system log features in log analysis: A survey," *Appl. Sci.*, vol. 13, no. 8, p. 4930, Apr. 2023.

[236] K. Yin, M. Yan, L. Xu, Z. Xu, Z. Li, D. Yang, and X. Zhang, "Improving log-based anomaly detection with component-aware analysis," in *Proc. IEEE Int. Conf. Softw. Maintenance Evol. (ICSME)*, Sep. 2020, pp. 667–671.

[237] S. Badreddine, A. d'Avila Garcez, L. Serafini, and M. Spranger, "Logic tensor networks," *Artif. Intell.*, vol. 303, Feb. 2022, Art. no. 103649.

[238] B. Zhang, H. Zhang, P. Moscato, and A. Zhang, "Anomaly detection via mining numerical workflow relations from logs," in *Proc. Int. Symp. Reliable Distrib. Syst. (SRDS)*, Sep. 2020, pp. 195–204.

[239] L. Xiang, X. Jin, L. Yi, and G. Ding, "Adaptive region embedding for text classification," in *Proc. 33rd AAAI Conf. Artif. Intell., AAAI, 31st Innov. Appl. Artif. Intell. Conf., IAAI 9th AAAI Symp. Educ. Adv. Artif. Intell. (EAAI)*, 2019, pp. 7314–7321.

[240] Y. Xie, L. Ji, and X. Cheng, "An attention-based GRU network for anomaly detection from system logs," *IEICE Trans. Inf. Syst.*, vol. E103.D, no. 8, pp. 1916–1919, 2020.

[241] P. Cheansunan and P. Phunchongharn, "Detecting anomalous events on distributed systems using convolutional neural networks," in *Proc. IEEE 10th Int. Conf. Awareness Sci. Technol. (iCAST)*, Oct. 2019, pp. 1–5.

[242] J. Liu, J. Li, and C. Wu, "An efficient massive log discriminative algorithm for anomaly detection in cloud," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, Dec. 2019, pp. 1–6.

[243] S. Huang, Y. Liu, C. Fung, R. He, Y. Zhao, H. Yang, and Z. Luan, "HitAnomaly: Hierarchical transformers for anomaly detection in system log," *IEEE Trans. Netw. Service Manage.*, vol. 17, no. 4, pp. 2064–2076, Dec. 2020.

[244] B. Zhu, J. Li, R. Gu, and L. Wang, "An approach to cloud platform log anomaly detection based on natural language processing and LSTM," in *Proc. 3rd Int. Conf. Algorithms, Comput. Artif. Intell.* New York, NY, USA: Association for Computing Machinery, Dec. 2020, pp. 1–7.

[245] C. Xiao, J. Huang, and W. Wu, "Detecting anomalies in cluster system using hybrid deep learning model," in *Parallel Architectures, Algorithms and Programming*, H. Shen and Y. Sang, Eds. Singapore: Springer, 2020, pp. 393–404.

[246] B. Dit, L. Guerrouj, D. Poshyvanyk, and G. Antoniol, "Can better identifier splitting techniques help feature location?" in *Proc. IEEE 19th Int. Conf. Program Comprehension*, Jun. 2011, pp. 11–20.

[247] V.-H. Le and H. Zhang, "Log-based anomaly detection with deep learning: How far are we?" in *Proc. IEEE/ACM 44th Int. Conf. Softw. Eng. (ICSE)*, May 2022, pp. 1356–1367.

[248] S. Lupton, H. Washizaki, N. Yoshioka, and Y. Fukazawa, "Log drift impact on online anomaly detection workflows," in *Product-Focused Software Process Improvement*, R. Kadgien, A. Jedlitschka, A. Janes, V. Lenarduzzi, and X. Li, Eds. Cham: Springer Nature Switzerland, 2024, pp. 267–283.

[249] M. Du, Z. Chen, C. Liu, R. Oak, and D. Song, "Lifelong anomaly detection through unlearning," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur.* New York, NY, USA: Association for Computing Machinery, Nov. 2019, pp. 1283–1297, doi: 10.1145/3319535.3363226.

[250] J. Antic, J. P. Costa, A. Cernivec, M. Cankar, T. Martincic, A. Potocnik, G. B. Elguezabal, N. Leligou, and I. T. Boigues, "Runtime security monitoring by an interplay between rule matching and deep learning-based anomaly detection on logs," in *Proc. 19th Int. Conf. Design Reliable Commun. Netw. (DRCN)*, Apr. 2023, pp. 1–5.

[251] S. He, X. Zhang, P. He, Y. Xu, L. Li, Y. Kang, M. Ma, Y. Wei, Y. Dang, S. Rajmohan, and Q. Lin, "An empirical study of log analysis at Microsoft," in *Proc. 30th ACM Joint Eur. Softw. Eng. Conf. Symp. Found. Softw. Eng.*, Nov. 2022, pp. 1465–1476.

[252] T. Wittkopp and A. Acker, "Decentralized federated learning preserves model and data privacy," in *Proc. Int. Conf. Service-Oriented Comput.*, H. Hacid, F. Outay, H.-Y. Paik, A. Alloum, M. Petrocchi, M. R. Bouadjenek, A. Beheshti, X. Liu, and A. Maaradji, Eds. Cham, Switzerland: Springer, 2021, pp. 176–187.

[253] R. Chen, S. Zhang, D. Li, Y. Zhang, F. Guo, W. Meng, D. Pei, Y. Zhang, X. Chen, and Y. Liu, "LogTransfer: Cross-system log anomaly detection for software systems with transfer learning," in *Proc. IEEE 31st Int. Symp. Softw. Rel. Eng. (ISSRE)*, Oct. 2020, pp. 37–47.

[254] L. Ruff, N. Görnitz, L. Deecke, S. A. Siddiqui, R. A. Vandermeulen, A. Binder, E. Müller, and M. Kloft, "Deep one-class classification," in *Proc. ICML*, 2018, pp. 4393–4402.

[255] S. Huang, Y. Liu, C. Fung, R. He, Y. Zhao, H. Yang, and Z. Luan, "Transfer log-based anomaly detection with pseudo labels," in *Proc. 16th Int. Conf. Netw. Service Manage. (CNSM)*, Nov. 2020, pp. 1–5.

[256] T. Marlaithong, V. C. Barroso, and P. Phunchongharn, "A hyperparameter tuning approach for an online log parser," in *Proc. 18th Int. Conf. Electr. Eng./Electron., Comput., Telecommun. Inf. Technol. (ECTI-CON)*, May 2021, pp. 1036–1040.

[257] J. Zhaoxue, L. Tong, Z. Zhenguo, G. Jingguo, Y. Junling, and L. Liangxiong, "A survey on log research of AIOps: Methods and trends," *Mobile Netw. Appl.*, vol. 26, no. 6, pp. 2353–2364, Dec. 2021.

[258] S. He, P. He, Z. Chen, T. Yang, Y. Su, and M. R. Lyu, "A survey on automated log analysis for reliability engineering," *ACM Comput. Surv.*, vol. 54, no. 6, pp. 1–37, Jul. 2022.

SCOTT LUPTON is currently pursuing the Ph.D. degree with the Department of Computer Science and Engineering, Waseda University, Tokyo. He is also the Head of the Data Science Initiative (DSI) and Technology Innovation Group (TIG), Nomura Securities Company Ltd. His research interests include machine learning, anomaly detection, systems support, and production reliability.

HIRONORI WASHIZAKI (Member, IEEE) received the Ph.D. degree in information science from Waseda University, Tokyo, in 2003. He is currently a Professor and the Associate Dean of the Research Promotion Division, Waseda University. He is also a Visiting Professor with the National Institute of Informatics and an Advisor with the University of Human Environments. He works as the Outside Director at eXmotion Company Ltd. and an Advisor at System Information Company Ltd. He spearheads the evolution project of the Guide to the Software Engineering Body of Knowledge (SWEBOK). He has been the lead on a large-scale professional training and education program SmartSE, which encompasses the IoT, AI, software engineering, and business. His research interests include systems and software engineering, machine learning software engineering, and ICT education. He has served as the IEEE Computer Society President-Elect 2024 and the President 2025. He has also served as a Convener of ISO/IEC/JTC1 SC7/WG20, the Chair of IPSJ SIGSE, and the Chair of JUSE SQiP. He is an Associate Editor of IEEE Transactions on Emerging Topics in Computing, Steering Committee Member of CSEE&T and APSEC, and an Advisory Committee Member of COMPSAC. He is a Professional Member of IEEE-Eta Kappa Nu.

NOBUKAZU YOSHIOKA (Member, IEEE) received the B.E. degree in electronic and information engineering from Toyama University, in 1993, and the M.E. and Ph.D. degrees from the School of Information Science, Japan Advanced Institute of Science and Technology, in 1995 and 1998, respectively. From 1998 to 2002, he was with Toshiba Corporation, Japan. From 2002 to 2004, he was a Researcher, and from 2004 to 2021, he was an Associate Professor with the National Institute of Informatics, Japan. From 2021 to 2024, he was a Professor with Waseda Research Institute for Science and Engineering, Waseda University, Japan. He is currently the CEO of QAML Inc. and a Guest Professor of the Research Institute for Science and Engineering, Waseda University. His research interests include security and privacy software engineering and software engineering for machine learning-based systems.

YOSHIAKI FUKAZAWA (Member, IEEE) received the B.E., M.E., and D.E. degrees in electrical engineering from Waseda University, Tokyo, Japan, in 1976, 1978, and 1986, respectively. He is currently a Professor with the University of Human Environment. His research interests include software engineering, especially the reuse of object-oriented software.