Practitioners' Expectations on Log Anomaly Detection

Xiaoxue Ma, Yishu Li, Jacky Keung, Xiao Yu, Huiqi Zou, Zhen Yang, Federica Sarro and Earl T. Barr

***Abstract*—Log anomaly detection has become a common practice for software engineers to analyze software system behavior. Despite significant research efforts in log anomaly detection over the past decade, it remains unclear what are practitioners' expectations on log anomaly detection and whether current research meets their needs. To fill this gap, we conduct an empirical study, surveying 312 practitioners from 36 countries about their expectations on log anomaly detection. In particular, we investigate various factors influencing practitioners' willingness to adopt log anomaly detection tools. We then perform a literature review on log anomaly detection, focusing on publications in premier venues from 2015 to 2025, to compare practitioners' needs with the current state of research. Based on this comparison, we highlight the directions for researchers to focus on to develop log anomaly detection techniques that better meet practitioners' expectations.**

***Index Terms*—Automated log anomaly detection, empirical study, practitioners' expectations**

I. Introduction

LOGS are a crucial source of information in software systems, capturing system runtime behavior and aiding in software engineering tasks like program comprehension [1], [2], anomaly detection [3]–[6], and failure diagnosis [7], [8]. Extensive machine learning (ML) and deep learning (DL)-based techniques have been proposed in research for automated log anomaly detection [9]–[11], with the aim of reducing manual analysis costs. Concurrently, industrial log monitoring tools have emerged to provide log analysis services, including log anomaly detection. However, no prior studies have investigated practitioners' expectations of these techniques or tools. It remains unclear whether practitioners appreciate the current log anomaly detection techniques or log monitoring tools, the factors influencing their decisions to adopt them, and their minimum thresholds for adoption. Gaining insights from practitioners is essential to identify critical issues and guide researchers in developing solutions tailored to meet the needs of practitioners.

In this paper, we follow a mixed-methods approach to gain insights into practitioners' expectations on log anomaly detection: 1) Initially, we conduct a series of semi-structured interviews with 15 professionals with an average of 8.07 years of software development/maintenance experience. Through these interviews, we investigate the current use of log monitoring tools, the challenges associated with them, the perceived importance of automated log anomaly detection tools, and practitioners' expectations for these automated tools. 2) We then proceed with an exploratory survey involving 312 software practitioners from 36 countries. This quantitative approach allows us to validate practitioners' expectations uncovered in our interviews on a broader scale. 3) Finally, we perform a comprehensive literature review on log anomaly detection spanning from 2015 to 2025, encompassing the last decade. We scrutinize research papers published in premier venues during this period, comparing these proposed techniques against the criteria that practitioners have for adoption.

We address the following four Research Questions (RQs):

RQ1: What is the state of log monitoring tools, and what are the issues? The primary reasons surveyed practitioners abstain from using these tools include concerns or doubts about these tools, leading them to rely on manual analysis instead, and a lack of awareness regarding the existence of these tools. About half of those with experience using these tools express dissatisfaction, citing issues such as "The tool requires compatibility with different platforms and technologies", "The tool cannot provide a rationale for why a log is labeled as an anomaly", and "The tool cannot efficiently analyze large volumes of log data while maintaining effectiveness and efficiency". Additionally, one-third of practitioners indicate such tools are unable to automatically detect log anomalies. Furthermore, over 74% of practitioners indicate that data resources such as historical labeled log data, metrics (system performance indicators), and traces (records of request journeys through a system) are sometimes or always available for log anomaly detection.

RQ2: Are automated log anomaly detection tools important for practitioners? 95.5% of surveyed practitioners consider such automated tools to be essential or worthwhile for their software maintenance. Furthermore, many believe that user-friendly automated tools could enhance the effectiveness and efficiency of log anomaly detection, thus reducing the need for manual effort.

RQ3: What are practitioners' expectations of automated log anomaly detection tools? In the research area, there are

Xiaoxue Ma and Yishu Li are with the Department of Electronic Engineering and Computer Science, Hong Kong Metropolitan University, Hong Kong 999077, China. E-mail: kxma@hkmu.edu.hk, sliy@hkmu.edu.hk.

Jacky Keung is with the Department of Computer Science, City University of Hong Kong, Hong Kong 999077, China. E-mail: jacky.keung@cityu.edu.hk.

Xiao Yu is with the State Key Laboratory of Blockchain and Data Security, Zhejiang University, Hangzhou 310058, China. E-mail: xiao.yu@zju.edu.cn.

Huiqi Zou is with the Department of Computer Science, Johns Hopkins University, Baltimore, MD 21218 USA. E-mail: hzou11@jh.edu.

Zhen Yang is with the Shandong University, Qingdao 266237, China. E-mail: zhenyang@sdu.edu.cn.

Federica Sarro and Earl T. Barr are with the Department of Computer Science, University College London, WC1E 6BT London, U.K. E-mail: f.sarro@ucl.ac.uk, e.barr@ucl.ac.uk.

Corresponding author: Xiao Yu.

Digital Object Identifier 10.1109/TSE.2025.XXXXXXXX

This article has been accepted for publication in IEEE Transactions on Software Engineering. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TSE.2025.3586700

two primary granularities for log anomaly detection: log event level (a single log) and log sequence level (a log sequence consisting of multiple logs). Among the surveyed practitioners, 70.5% tend to perform log sequence level analysis, with their first preference being to group log sequences according to window sizes. Practitioners consider recall (identifying real anomalies), precision (accuracy of identified anomalies), and the efficiency of real-time anomaly detection as the most crucial evaluation metrics influencing their acceptance of these tools. Over 70% expect recall and precision above 60%. In addition, more than 78% would consider using automated log anomaly detection tools if they could customize them to process diverse log structures and provide explanations for detected anomalies. Moreover, more than half of practitioners expect these tools to handle at least 100,000 logs, with installation, configuration, and learning taking no more than an hour. They also prefer anomalies to be identified within 5 seconds of their appearance.

RQ4: How close are the current state-of-the-art log anomaly detection studies to satisfying practitioners' needs before adoption? We identify 47 papers from premier venues (encompassing various tracks) on log anomaly detection from 2015 to 2025 and explore the gap between proposed techniques and practitioners' expectations across nine aspects.

Our main findings include: (1) Only 4 incorporate metrics or traces to aid in anomaly detection, despite 83.7% and 74.9% of practitioners indicating the availability of these data types. (2) Only 6 papers explicitly perform log event level anomaly detection, which is the second preference for practitioners. (3) Over half of the surveyed studies do not mention the log anomaly detection time, despite a majority advocating for real-time anomaly detection. (4) Few or no studies address handling logs with diverse structures, providing a rationale for detected anomalies, customization, or privacy protection, all of which are significant concerns for surveyed practitioners. Despite most studies showcasing accurate anomaly detection on public datasets, half of the practitioners still refrain from adopting the proposed techniques due to main concerns over their interpretability, user-friendliness, and ability to handle various log data (details in Section III-D). This highlights a significant need for improvements to better meet practitioners' needs.

In summary, our work makes the following contributions:

(1) We interviewed 15 professionals and surveyed 312 practitioners from 36 countries to gain insights into their expectations. This includes their perspectives on the current log monitoring tools, the importance of automated log anomaly detection tools, the factors influencing their adoption of these tools, and their minimum adoption thresholds.

(2) We conduct an extensive literature review, following inclusion criteria and three steps of examination to identify relevant papers published in premier venues over the past decade. We then compare the current state of research with practitioners' expectations.

(3) We highlight potential implications to align research efforts with the needs of practitioners and propose actionable strategies for further enhancement. Furthermore, we validate our implications through a follow-up survey, reinforcing the robustness and practical relevance of our conclusions.

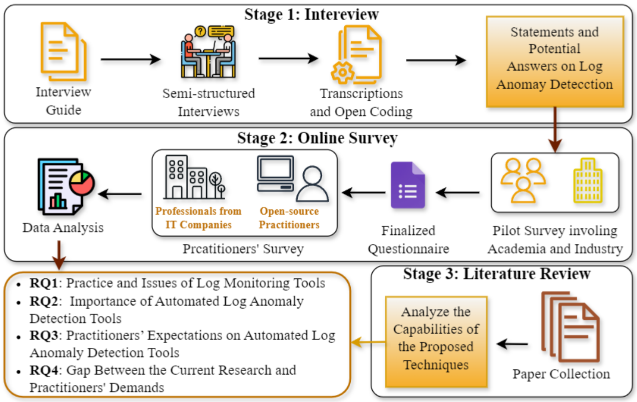

Stage 1: Intereview
Interview Guide
Semi-structured Interviews
Transcriptions and Open Coding
Statements and Potential Answers on Log Anomay Detecction
Stage 2: Online Survey
Data Analysis
Professionals from IT Companies
Open-source Practitioners
Prcatitioners' Survey
Finalized Questionnaire
Pilot Survey involing Academia and Industry
RQ1: Practice and Issues of Log Monitoring Tools
RQ2: Importance of Automated Log Anomaly Detection Tools
RQ3: Practitioners' Expectations on Automated Log Anomaly Detection Tools
RQ4: Gap Between the Current Research and Practitioners' Demands
Stage 3: Literature Review
Analyze the Capabilities of the Proposed Techniques
Paper Collection

Fig. 1: The overview of the research methodology.

II. Research Methodology

Our research methodology follows a mixed-methods approach [12], as illustrated in Figure 1, comprising three main stages. ***Stage* 1**: Interview with professionals to explore their practices in detecting log anomalies, their experiences, the issues they have encountered when using log monitoring tools for log anomaly detection, and their expectations on log anomaly detection. ***Stage* 2**: Perform an online survey designed to validate and expand upon the findings derived from the interviews regarding log anomaly detection. ***Stage* 3**: Conduct a comprehensive literature review to analyze whether or to what extent current state-of-the-art log anomaly detection techniques have fulfilled practitioners' needs and expectations. Both the interviews and surveys conducted in this study adhere to the guidelines set forth by the relevant institutional review board.

A. Stage 1: Interview

Protocol. We conduct a series of semi-structured interviews, utilizing both video calls and face-to-face discussions, to comprehensively examine practitioners' practices, issues, and expectations regarding log anomaly detection. We generate our interview questions from two sources: first principles and standard open-ended interviews [13]. Initially, we draft the questions based on our experience and knowledge, which are then validated through exploratory interviews with 15 professionals. This initial process results in a final set of 10 main questions, which remains unchanged throughout the following interview process. We ceased inviting additional professionals after conducting 15 interviews because we reached data saturation, meaning no new significant insights emerged, indicating comprehensive coverage of the relevant aspects. Before each interview, interviewees are provided with a brief introduction to the study's background information, ensuring they are informed about the recording of the interview and emphasizing that their identities will be protected. Each interview lasts between 40 to 60 minutes, and the first author presents to document the transcript and ask follow-up questions based on the interviewees' responses for a more in-depth understanding.

In each interview, we first pose demographic questions to the interviewees to gather information about their background, including job roles and work experience. Subsequently, we explore their practices with log monitoring tools for log anomaly detection and the issues they encountered. Finally, we ask open-ended questions to gather their perceptions and expectations regarding log anomaly detection.

Interviewees. For interviews, we employ expert sampling, a subtype of purposive sampling [14], to target professionals with relevant expertise in log analysis to gather valuable insights. We invited 15 interviewees from eight IT companies worldwide, including Microsoft, Google, Alibaba, Intel, Huawei, etc. These interviewees are professionals from various roles closely engaged in log analysis within software development and maintenance, such as developers, operation and maintenance engineers, and others. The IT project experience of our interviewees varies from 4 years to 16 years, with an average professional experience of 8.07 years (minimum: 4, median: 7, maximum: 16, standard deviation: 3.45 years). During the interviews, we systematically examined key log usage scenarios to inform the design of our subsequent online survey. Our interview analysis identified four distinct but interconnected scenarios for log usage in industry practice. (1) Development debugging: Engineers leverage logs during software development to trace variable values and program execution flows, which significantly reduces debugging time. (2) Production diagnostics: In live production environments (i.e., where a company's software applications and services operate to serve real users), traditional debugging tools are often unavailable. In these cases, logs become the primary means of identifying anomalies, allowing teams to analyze logs to pinpoint issues that would otherwise require time-consuming replication of the environment. (3) Operational monitoring: For system operations, logs provide real-time visibility into system health, enabling teams to detect and escalate critical incidents promptly. (4) Data analytics: Beyond troubleshooting, processed logs serve as valuable data sources for user behavior analysis and recommendation systems, as demonstrated by e-commerce platforms that transform user activity logs into personalized shopping recommendations.

Open Coding Analysis. The first author transcribed the interviews and performed open coding [15] to generate an initial set of codes. The second author then verified the codes and provided suggestions for improvement. After incorporating the suggestions, the two authors independently analyzed and sorted the opinion cards into potential descriptions for the questionnaire. The Cohen's Kappa value between the two authors is 0.72, indicating a substantial agreement. Any disagreements have been discussed to reach a consensus. To mitigate the bias of the two authors in sorting descriptions, another two authors have also reviewed and confirmed the final set of survey descriptions. Ultimately, based on the results of the interviews, we identified eight issues with log monitoring tools for log anomaly detection, eight reasons for not using them, five degrees of importance regarding automated log anomaly detection tools, and nine expectations from these tools.

B. Stage 2: Online Survey

1) Survey Design: We design our survey through a two-step process: insights from interviews and a pilot survey. Initially, based on insights gained from the interviews, we observed that interviewees' experiences with log monitoring tools significantly influenced their responses. Consequently, we implement a branching mechanism in the survey to tailor questions according to their specific experiences, resulting in a finalized 24-question survey. Subsequently, we conduct a pilot survey to refine the survey description and question types, ensuring clarity and appropriateness for the target respondents.

Based on the insights from the interviews, the survey comprises various question types, including single-choice, multiple-choice, rating questions on a 5-point Likert scale, ranging from "Strongly Disagree" to "Strongly Agree", and short answer questions. Additionally, we include an "I don't know" option for surveyed practitioners who do not understand our descriptions. To mitigate bias resulting from practitioners' unfamiliarity with log anomaly detection, we provide a detailed explanation of its workflow and user scenarios.

(1) Demographics. This section collects demographic information of the surveyed practitioners, including their countries of residence, primary job roles, professional experiences and programming languages, and team sizes.

(2) Practices and Issues of Log Monitoring Tools for Log Anomaly Detection. This section first offers surveyed practitioners a concise overview of log monitoring tools for detecting log anomalies. Then, it explores the specific tools they have employed and the issues they have met during usage for log anomaly detection. For practitioners who have not utilized such tools, we inquire about their primary reasons. In addition, we investigate the availability of data resources to aid in log anomaly detection.

(3) Importance of Automated Log Anomaly Detection Tools. Given practitioners' potential doubts and concerns about existing log monitoring tools for log anomaly detection, or their possible unawareness of such tools, this section assesses the importance of an automated log anomaly detection tool, assuming it can meet practitioners' expectations. Practitioners are prompted with the question: "If we were to build an automated log anomaly detection tool that fully satisfies all your expectations, do you consider this tool important in your system maintenance?". They are asked to evaluate the importance of the automated tool using statements such as "Essential" (I will use this tool daily), "Worthwhile" (I will use this tool), "Unimportant" (I will not use this tool), or "Unwise" (This tool will have a negative impact on log anomaly detection for me or my team). Accordingly, practitioners are queried about their primary motivations for using or reasons for not using the automated tool.

(4) Practitioners' Expectations on Automated Log Anomaly Detection Tools. This section investigates surveyed practitioners' expectations regarding the detection granularity levels for utilizing automated log anomaly detection tools, including log event level (detecting a single log) and log sequence level (detecting a log sequence containing multiple logs). Strategies for grouping logs into log sequences include grouping logs based on window sizes (e.g., 2-20 logs per sequence), fixed time intervals (e.g., logs within every 5 minutes per sequence), and timestamps (logs grouped at a specific moment or instant) or sessions (e.g., block_id). The section then explores the key factors influencing their acceptance of these tools for software system maintenance, such as *correct detection rates* (whether the real anomalies can be correctly identified by the tools and whether the identified anomalies by the tools are real anomalies), *real-time detection* (whether abnormal logs are detected in real-time as they occur), *interpretability* (whether the tool can provide a rationale for why a detected log is labeled as an anomaly), *generalizability* (whether the tool can analyze logs with various structures), *customization* (whether the tool can be easily adjusted to meet different requirements, like adding new algorithms or adapting thresholds for defining anomalies), *easy to use* (whether the tool is easy to install, configure, and utilize), and *security and privacy measures* (whether privacy and security measures are implemented and stated in the tools for sensitive information protection). Furthermore, the section explores the minimum adoption threshold of such tools in terms of effectiveness (measured by recall and precision values), efficiency (time required to detect anomalies and time from the tool installation to successful utilization), scalability (capacity to handle a specified number of logs), and privacy protection (required measures provided by the tool).

Considering many DL-based techniques in academia have been proposed for log anomaly detection and have demonstrated strong performance (i.e., recall and precision over 80%) on public datasets, we ask practitioners about their willingness to use such techniques. Finally, practitioners are invited to provide free-text comments regarding automated log anomaly detection and our survey. In particular, we inquire about potential opportunities for leveraging large language models (LLMs) like ChatGPT to enhance log anomaly detection. Practitioners may or may not choose to provide final comments.

Before launching the survey, we conducted a pilot survey involving three industrial experts and two academics specializing in software operation and maintenance research, none of whom are interviewees or surveyed practitioners. This pilot aims to gather feedback on the survey's length, clarity, and the understandability of terms used. Based on their insights, we made minor adjustments to the draft survey, refining it into a finalized version. To cater to a global audience, we offer an English version of the survey through Google Form [16], with the full text publicly accessible [17]. Additionally, we translate the survey into Chinese and host it on a popular survey platform in China [18] to ensure accessibility for practitioners in the region.

2) Participant Recruitment: The survey was conducted online. Given the unknown total population of log analysis experts, it was challenging to determine an appropriate sample size for data collection and to identify practitioners with practical experience in log anomaly detection who were willing to participate in the survey. Therefore, in accordance with previous research methodologies [19], we utilized purposive sampling and manually screened professional profiles worldwide and purposely selected practitioners who had been involved in the log-related analysis processes of software systems. Potential respondents were primarily sought on platforms such as GitHub and within academic research groups. For instance, we mined commit logs in GitHub repositories to identify the email addresses of developers or engineers who had actively contributed to log-related projects. After reviewing their online profiles to verify their expertise and relevance to our study, we contacted these individuals to confirm their willingness to participate in our survey. This initial contact included a detailed explanation of the survey's objectives and the nature of the information to be collected, explicitly stating that no sensitive information would be required. If the potential participants responded affirmatively, we formally issued an invitation letter along with the survey links. To gather a diverse pool of surveyed practitioners, we also engaged professionals within our social and professional circles employed at various IT companies. We requested their support in sharing our survey with their colleagues. Invitations were extended to contacts at prominent companies such as Google, Microsoft, Intel, ByteDance, Cisco, Alibaba, Huawei, and others. In total, we received 312 valid responses. These responses originate from 36 countries, with China, the United States, and Germany among the top three regions with the highest number of surveyed practitioners. Most surveyed practitioners actively engage in software development (40%) and operation and maintenance (29%), with 3-10 years of experience in medium-sized project teams comprising 6 to 40 members. An overview of the distribution of surveyed practitioners' roles and their experiences is detailed in Table I.

TABLE I: The distribution of practitioners' experiences.

Role	#	Experience			Team Size (ppl.)		
		<3y	3-10y	>10y	1-5	6-40	>40
Software Development	125	51	83	31	45	96	24
Operations & Maintenance	90	29	17	4	26	21	3
Software Testing	31	7	19	5	11	17	3
Architect	23	5	14	4	1	12	10
Algorithm Design	23	7	13	3	8	12	3
Project Management	8	-	5	3	2	5	1
Others	12	5	5	2	3	7	2

3) Result Analysis: We analyze the survey results based on question types. For single-choice and multiple-choice questions, we report the percentage of each selected option. Open-ended questions undergo qualitative analysis through careful examination of responses. We draw bar charts to highlight possible trends in the Likert-scale answers. For the open-ended questions, the first two authors independently analyzed these, categorizing them into specific characteristics. We exclude "I don't know" ratings since they constitute a small minority, i.e., less than 1% of all valid responses.

C. Stage 3: Literature Review

First, we use IEEE, ACM, DBLP, and Google Scholar to search for publications from 2015 to 2025 by using the query: ("log"|| "log anomaly") & ("detect/detection" || "analyze/analysis" || "predict/prediction" || "classify/classification"). Then, we keep those papers published in premier peer-reviewed venues[1], yielding 88 papers. These papers span various tracks, with the majority originating from the research and technical tracks, while the remainder comes from the industry track, workshops, and practical track. From 88 papers, we then filter out irrelevant publications according to the following **inclusion criteria**: 1) The publication should propose a new log anomaly detection technique, and 2) the publication predicts a binary outcome (i.e., normal or abnormal log). To assess whether the publications satisfy our inclusion criteria, we manually examine each publication following the three steps adopted in previous surveys [20], [21]: 1) **Title:** if the publication's title clearly does not match our inclusion criteria, then it is excluded; 2) **Abstract:** if the publication's abstract does not meet our inclusion criteria, then it is excluded; and 3) **Body:** if the examined publication neither satisfies the inclusion criteria nor contributes to this survey, then it is excluded. Among the 88 articles, 35 meet the inclusion criteria. Starting from these 35 articles, we perform one level of forward snowballing [22] and gather 71 additional publications. We assess these additional publications by using the same inclusion criteria and process described above, identifying 12 articles that meet the criteria. Ultimately, we identify a total of 47 papers: 8 from ISSRE, 5 from ICSE, 3 from ESEC/FSE, 3 from ASE, 2 from KDD, 2 from ICWS, 1 from ICPC, 1 from AAAI, 1 from ICSE/SEIP, 1 from ICDE, 1 from CCS, 1 from ICDM, 1 from ICDMW, 1 from ACSAC, 1 from ICSME, 4 from TNSM, 3 from ASEJ, 2 from TSC, 1 from TSE, 1 from TOSEM, 1 from TC, 1 from TDSC, 1 from IJCAI, and 1 from ICT Express.

For each log anomaly detection paper, two authors read its content and analyze the capabilities of the proposed technique. For instance, Guo et al. [23] utilize a Transformer-based framework, combining pre-training on the source domain and adapter-based tuning on the target domain, and evaluate it on three datasets with different log grouping strategies. Then we infer that they focus on log sequence level anomaly detection, supporting cross-project generalizability. We assess effectiveness and efficiency satisfaction rates using the lowest precision/recall values and the maximum detection time provided. The first two authors discuss any differences in their capability analysis and confirm the final results through further reading of the papers.

III. Results

A. RQ1: Practices and Issues of Log Monitoring Tools

In this research question, we investigate practitioners' practices regarding log monitoring tools, focusing on their tool preferences for log anomaly detection, reasons for non-utilization, data source availability, and encountered issues during tool usage for log anomaly detection.

1) Practices of log monitoring tools: According to survey results, 174 out of 312 (55.8%) practitioners report using log monitoring tools for log anomaly detection, while the remaining 138 indicate they do not use these tools. Figure 2 illustrates the percentage of usage of various tools for log anomaly detection. Practitioners may use more than one tool, resulting in a total percentage exceeding 100%. The results reveal that 36.8% of practitioners use *Elastic*, and 31.6% utilize *Amazon CloudWatch Logs*. Following tools, *Ali Cloud Simple Log Service*, *DataDog*, *LogicMonitor*, and *Nagios* rank as the third to sixth most commonly used tools, with adoption rates of 17.2%, 16.7%, 15.5%, and 12.1%, respectively. Less prevalent tools, including *Graylog*, *Sentry*, *Digilogs*, *Huawei Cloud Log Tank Service*, and *Tencent Cloud Log Service*, have adoption rates below 10%. Tools with usage rates under 3%, such as *Better Stack*, *LogCat*, and *LogDNA*, along with tools specified by practitioners (e.g., internal systems), are categorized as *Others*, with a total usage rate of 22.3%.

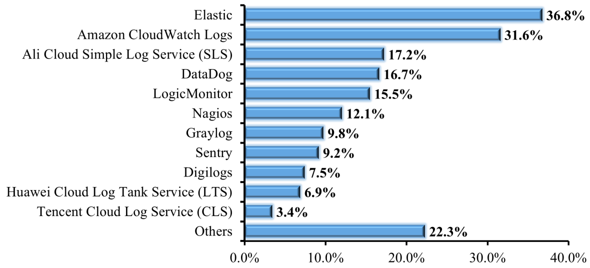


Fig. 2: The percentages of tool usage for log anomaly detection.

Among practitioners who have used log monitoring tools, one-third of practitioners report that the tools cannot automatically identify log anomalies and require manual analysis. We further investigate their awareness of the underlying techniques of these tools. The results indicate that 75.5% of tools utilize keyword-based heuristic methods, while 26.3% of tools are developed with ML or DL techniques. Some tools incorporate intersecting techniques, leading to a total percentage exceeding 100%.

> **Finding 1.** Over half (55.8%) of practitioners use log monitoring tools for log anomaly detection, with *Elastic* and *Amazon CloudWatch Logs* being the top choices. Notably, one-third find these tools incapable of automated anomaly detection.

For the 138 practitioners who have not used log monitoring tools for log anomaly detection, the primary reasons cited are that manual analysis suffices for detecting abnormal logs (49.3%) and a lack of awareness of existing tools (47.1%). Additionally, other reasons mentioned due to the multiple-choice nature of the question include a lack of technical expertise to use the tools (29%), doubts about the effectiveness and reliability of the tools (21%), a lack of compatibility with different platforms and technologies (15.9%), time-consuming installation and learning process of the tools (13.8%), data security or user privacy concerns regarding using the tools (13%), expensive payment for using the tools (4%). As some of the surveyed practitioners noted:

✍ *Manual log analysis is straightforward and familiar to me. I haven't found an automated tool that can replicate our daily*

[1] We consider publications in conferences ranked A and A* according to the CORE Ranking (https://portal.core.edu.au/conf-ranks/) and journals falling in the Quartile Q1 according to the Journal Citation Reports (JCR) (https://jcr.clarivate.com/jcr/home).

practices as effectively as I'd like.

✎ *Our business is intricate, and we're skeptical about whether the automated tools can really adapt to our needs for finding actual anomalies. That's why we're still relying on manual analysis.*

✎ *I wasn't even aware that such tools existed. If I had known about them, I would definitely consider giving them a try, especially because the manual workload can be quite heavy.*

To summarize, apart from the unawareness of existing log monitoring tools, most practitioners still have concerns or doubts about the performance of these tools in multiple aspects.

Finding 2. Nearly half of the surveyed practitioners refrain from using log monitoring tools, primarily due to either their reliance on manual analysis, driven by concerns and doubts about these tools, or their unawareness of the existence of such tools.

Figure 3 illustrates the availability of data resources utilized by the surveyed practitioners in detecting log anomalies. They rate the availability of four specific data types (i.e., historical labeled normal logs, historical labeled abnormal logs, metrics, and traces) on a scale from "Always" to "Never". Metrics represent numerical data points indicating system performance indicators like response time, CPU usage, and memory consumption, while traces provide a comprehensive record of request journeys through a system, detailing timestamps, resource usage, and module interactions. The survey indicates that these resources are *always* available to 39.3%, 43.5%, 51.5%, and 45.2% of practitioners, respectively. Furthermore, 40.6%, 40.2%, 32.2%, and 29.7% report these data sources as *sometimes* available. Less than 20% of practitioners indicate that historical labeled normal and abnormal log data, as well as metrics, are *rarely* or *never* available for log anomaly detection. This suggests that while historical log data is available to more than 80% of practitioners, over 74% recognize the availability of other data types, such as metrics and traces, for log anomaly detection, as highlighted by one practitioner:

✎ *I usually use the metric and trace data because they contain more useful information for detecting log anomalies.*

Finding 3. At least 74% of the surveyed practitioners recognize the availability of historical labeled log data, metrics, and traces, with some indicating that these available data are helpful for log anomaly detection.

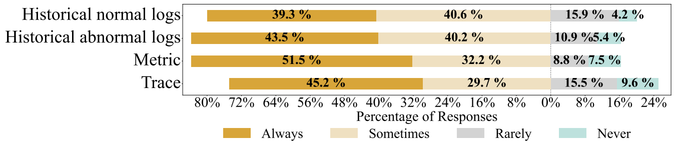


Fig. 3: Data resource availability for log anomaly detection.

2) Issues of log monitoring tools: Figure 4 demonstrates the issues encountered by surveyed practitioners when using log monitoring tools for log anomaly detection. Among practitioners utilizing these tools, 48.2% express dissatisfaction, and only 19.3% think these tools are satisfying. Over half of the practitioners (51%) consider compatibility to be the most significant issue. They present agreement that "the tool requires compatibility with different platforms and technologies". 48.7% of them consider "the tool cannot provide a rationale for why a log is labeled as an anomaly". Some practitioners share their opinions:

✎ *The tool didn't fit well with our system, making it challenging to incorporate it into our workflow.*

✎ *The tool is very simple. It just provides daily reports of anomalies that may be suspect and require much investigation to rule as a legitimate threat or nuisance scan.*

✎ *The anomalies detected lack sufficient details, making them less meaningful for our needs. That means we still rely on manual analysis to gather comprehensive information.*

Practitioners also express significant concern about the scalability of these tools. Specifically, 48.4% of the practitioners report that "the tool cannot efficiently analyze large volumes of log data while maintaining effectiveness and efficiency". These concerns are primarily voiced by practitioners working on medium and large-sized product development teams.

✎ *Sometimes, the tool is useful when handling logs during the development and testing phases. However, its accuracy and performance drop when dealing with logs from the production phase, where data usage is increasing.*

Another two main challenges pertain to the effectiveness of the log monitoring tools. A total of 47.8% of practitioners indicate that "the tools often fail to accurately detect real abnormal logs", whereas 44.7% complain "the tools often incorrectly identify the real normal log as anomalies". Some practitioners express doubts about the accuracy of these tools, leading to reduced reliance on them.

✎ *It's frustrating when the tool incorrectly detects anomalies because then we have to spend extra time doing manual analysis. Because of that, I find myself relying on it less for anomaly detection.*

Other issues highlighted by practitioners encompass concerns regarding privacy, usability, and efficiency. Notably, 44.1% express concerns regarding "the tool has the potential to disclose sensitive information", while 41% find "the installation and use of the tools are difficult". Additionally, 29.6% report that "the tools cannot detect anomalies or send alerts in time".

Finding 4. Practitioners highlight several notable issues with existing log monitoring tools for log anomaly detection. The majority (51%) express concerns about the compatibility of these tools with different platforms and technologies. Between 40% and 50% of practitioners raise concerns regarding various aspects of the tools, including their interpretability, scalability, effectiveness, privacy disclosure, and ease of use.

B. RQ2: The Importance of Automated Log Anomaly Detection Tools

In this research question, we investigate how practitioners evaluate the importance of automated log anomaly detection

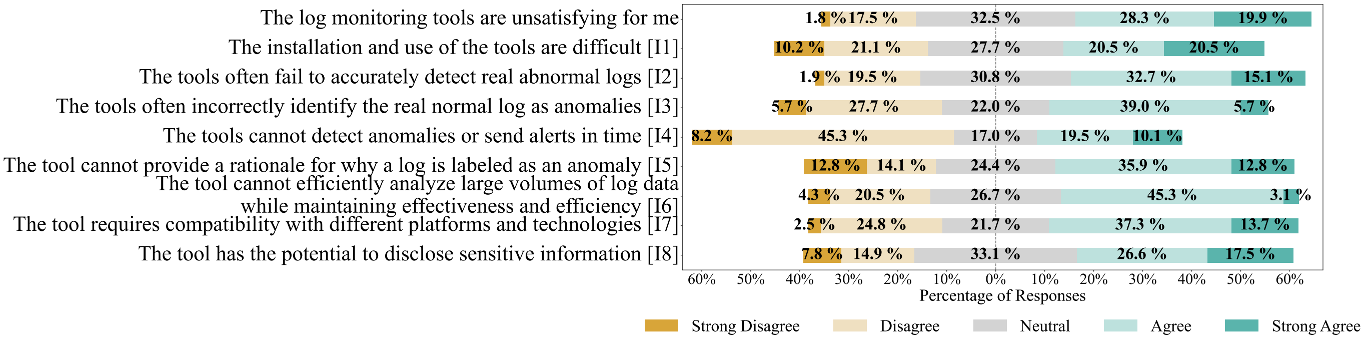


Fig. 4: The current issues with log monitoring tools for log anomaly detection.

tools, assuming their expectations are met. Figure 5 presents the ratings from surveyed practitioners. Overall, about 95.5% of surveyed practitioners regard automated log anomaly detection tools as either *essential* or *worthwhile*. Among practitioners who have experience with existing log monitoring tools, 97.7% state they would use this automated tool if it meets all their expectations. Additionally, 92.6% of those who have not previously used log monitoring tools express a willingness to adopt the automated log anomaly detection tool. Comments left by practitioners indicate that their primary motivation for using these tools is the reduction of manual analysis effort. As some practitioners who have used existing log monitoring tools suggested:

📝 *If the tool can accurately detect log anomalies and provide detailed information, it would really cut down the time we spend on investigating and analyzing issues.*

📝 *Having an automated tool is great; it really helps improve our work efficiency. I will definitely use it if the tool can provide accurate anomaly detection.*

A practitioner who has not used existing log monitoring tools remarked:

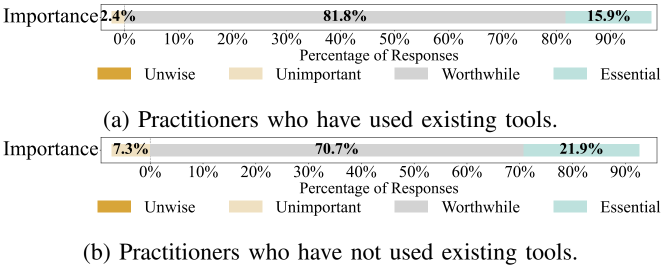
📝 *If the tool is compatible with our system, we will use it because it can help us reduce manual effort.*

A mere 2.4% of practitioners who have used existing log monitoring tools and 7.3% of practitioners who have not used such tools regard the automated log anomaly detection tool as *unimportant*, and indicate they would not use such tools. Their primary concerns are doubts about its effectiveness and a preference for internally developed tools over external ones.

📝 *I doubt that the tool can be developed to adapt to our complex system and be easy to use.*

📝 *Our product is designed for internal usage and will not be integrated with external log systems.*

Furthermore, to explore their attitudes toward the current research, we include a question at the end of our questionnaire asking whether participants would consider using log anomaly detection models proposed in research if they can achieve high recall and precision (>90%). In the question instructions, we provide a comprehensive introduction to the general log anomaly detection process along with a detailed workflow figure. We find that 31.1% of practitioners indicate they will not use these models. Specifically, 27.0% of practitioners who have used existing log monitoring tools indicate that they will not adopt such models. Their main concerns are that the models are difficult to use and may involve high computing costs. As some practitioners noted:



(a) Practitioners who have used existing tools.

(b) Practitioners who have not used existing tools.

Fig. 5: The importance of automated log anomaly detection tools.

📝 *A bit complicated. I want to use tools readily available quickly.*

📝 *Not necessarily, too complex.*

📝 *Do not consider, there is not enough computing power hardware equipment to support operation or additional expenses to access third-party services.*

For those who have not used existing log monitoring tools, 36.2% indicate that they will not adopt the proposed models from academia or do not regard them as a priority due to their unfamiliarity with the models and concerns about associated costs. Some comments reflecting their perspectives are shown below:

📝 *No. It requires expertise of model tuning.*

📝 *If it is a non-UI model, do not consider it. Because it is not very convenient to use.*

📝 *No, the cost of learning is too high.*

📝 *Not at the moment, cause I am not familiar with these models. I don't know how to fine-tune them or adjust their settings for the best results.*

Finding 5. The majority (95.5%) of practitioners view automated log anomaly detection tools as either essential or worthwhile for their practice. They believe that such automated tools can efficiently and accurately detect anomalies, thereby streamlining manual analysis efforts. In contrast, nearly one-third of practitioners indicate that they will not adopt the log anomaly detection models proposed in research.

C. RQ3: Practitioners' Expectations on Automated Log Anomaly Detection Tools

In this research question, we delve into practitioners' expectations regarding log anomaly detection, exploring aspects including *granularity level*, *evaluation metrics*, *effectiveness*, *efficiency*, *scalability*, and *privacy protection*.

Granularity level. From Figure 6, 70.5% of practitioners prefer detecting log anomalies at the sequence level, where an entire sequence is labeled as abnormal if any log within it is abnormal. Specifically, 30.1% group logs into sequences based on window sizes (e.g., every 20 logs), 24.0% use fixed time intervals (e.g., every 5 minutes), and 16.4% group logs by timestamp/session (e.g., block_id). Furthermore, 29.5% prefer log event level, making it the second most favored detection granularity among surveyed practitioners. Although we discuss with the interviewers to confirm that the options provided for practitioners are reasonable and comprehensive, we recognize that they may not cover all possible scenarios. Therefore, we include an "Other" option for the online survey. However, it is noteworthy that all responses from our practitioners fall within the predefined options. This indicates that the options may adequately reflect their experiences and needs.

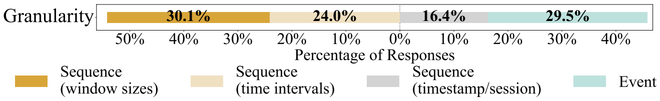


Fig. 6: The granularity of automated log anomaly detection tools.

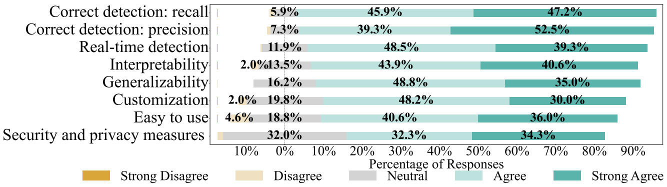


Fig. 7: The factors affecting practitioners' acceptance of using automated log anomaly detection tools.

Evaluation metrics. We analyze surveyed practitioners' opinions on evaluating automated log anomaly detection tools as shown in Figure 7. The findings reveal that 93.1% and 91.8% of practitioners agree that the ability to correctly identify real anomalies and the accuracy of the identified anomalies (***correct detection rates***) are the two most preferred evaluation metrics. As one practitioner noted: "*I want the tool to be able to detect potential anomalies and miss as few as possible*" while another stated: "*I only accept the tool if it can detect anomalies with high accuracy; otherwise, I'll have to manually judge, costing me time.*". Additionally, 87.8% of practitioners emphasize the importance of ***real-time detection*** for identifying anomalies. As a practitioner noted: "*I'm really looking for a real-time tool that can streamline the log anomaly detection process, making it much easier and ultimately boosting the detection efficiency.*". Furthermore, 84.5% value ***interpretability***, indicating that the tool should provide a rationale for why a log is labeled as an anomaly. ***Generalizability*** is crucial for 83.8% of practitioners, who think the tool should effectively analyze logs with diverse structures. As a practitioner emphasized: "*Detecting anomalies shouldn't always need specific knowledge of certain applications. It'd be great to automatically spot different log anomalies for wider use.*". ***Customization*** is also important, with 78.2% believing the tool should be easily adjustable to meet various requirements. For example, they want the ability to select different anomaly detection algorithms, configure specific log formats, and set custom alert thresholds. Moreover, 76.6% of practitioners consider ***easy to use*** as a significant factor for adopting the tool, with the question in the survey that asks whether it is easy to install and configure the tools, and learn how to use them. One practitioner also emphasizes that if the tool effectively fulfills its purpose, the time required for installation and configuration is acceptable: "*If a tool gets the job done, the installation and configuration process is just part of the deal. Honestly, overcoming a challenge during setup can even make the tool more interesting to use. What really matters is that it does what it's supposed to do.*". Lastly, 66.6% of practitioners agree that ***security and privacy measures*** should be implemented to protect sensitive information.

> **Finding 6.** More than 78% of surveyed practitioners consider using automated log anomaly detection tools if they can be customized to process logs with different structures and provide a rationale for the detected anomalies.

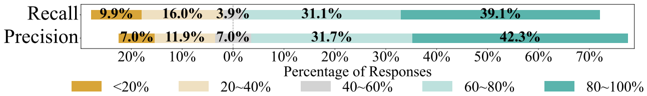


Fig. 8: The surveyed practitioners' satisfaction rate in terms of effectiveness.

Effectiveness. Figure 8 illustrates practitioners' satisfaction levels with automated log anomaly detection tools based on recall and precision values. The satisfaction rate is categorized into five capability ranges, from less than 20% to 80%-100%. Our survey indicates that 39.1% of practitioners will accept the tool only if the recall exceeds 80%, and 42.3% will do so if the precision exceeds 80%. Achieving recall and precision above 60% will satisfy 70.2% and 74.0% of practitioners, respectively.

> **Finding 7.** Correct detection rates (recall and precision) are the most critical factors influencing surveyed practitioners' acceptance of automated log anomaly detection tools. Over 70% of the surveyed practitioners expect the correct detection rates to exceed 60%.

Efficiency. For tool installation, configuration, and learning time (Figure 9), practitioners are most satisfied when these activities take less than 10 minutes or between 10-60 minutes, with 26.8% and 32.7% favoring these ranges, respectively. Satisfaction decreases as time increases, with 22.3% satisfied within 1-3 hours, 13.6% for 3-24 hours, and the least satisfaction at 4.5% for several days. Regarding anomaly detection

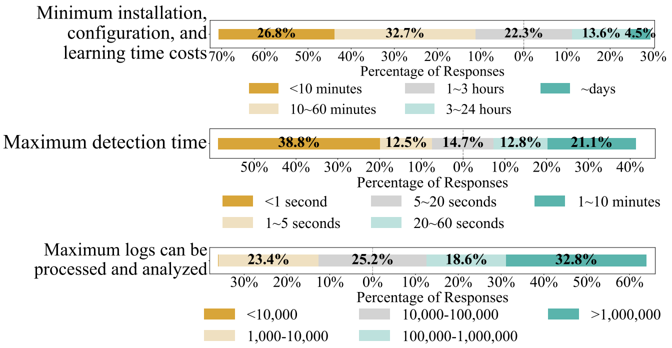


Fig. 9: The surveyed practitioners' satisfaction rate with various capability ranges in terms of efficiency and scalability.

time, most practitioners (around 80%) require detection to be within 1 minute. Specifically, 38.8% are only satisfied with detection in less than 1 second, 51.3% within 5 seconds, and 66% within 20 seconds. Interestingly, 21.1% are satisfied with detection taking longer (1-10 minutes). As one practitioner explained: "*Log anomalies do not necessarily need to be detected in real-time, and some anomalies may not cause program crashes immediately.*".

Scalability. Scalability explores the expectation of the maximum number of logs that can be processed and analyzed as the volume of logs grows while maintaining accurate and timely detection of anomalies. As shown in Figure 9, 23.4% of surveyed practitioners are satisfied with tools handling less than 10,000 logs. Satisfaction is 25.2% for tools managing between 10,000-100,000 logs and 18.6% for those handling 100,000-1,000,000 logs. The highest satisfaction, at 32.8%, is for tools capable of handling over 1,000,000 logs.

> **Finding 8.** More than half of the surveyed practitioners expect automated log anomaly detection tools to handle at least 100,000 logs, with installation, configuration, and learning time of less than 1 hour, and anomaly detection time to be under 5 seconds when an anomaly appears.

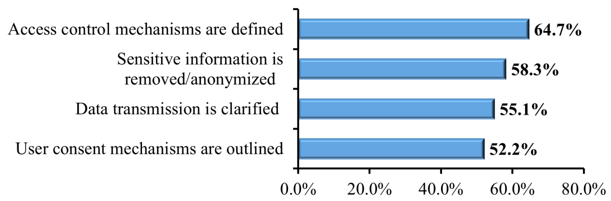


Fig. 10: The surveyed practitioners' satisfaction in terms of privacy protection.

Privacy protection. We explore the expectations of the surveyed practitioners on privacy protection. As shown in Figure 10, 64.7% of practitioners are satisfied with automated log anomaly detection tools that clearly define access control mechanisms. Additionally, more than half of the practitioners accept these tools if privacy protection measures are explicitly stated, such as removing or anonymizing sensitive information (58.3%), clarifying data transmission protocols like encryption (55.1%), and transparently outlining user consent mechanisms for data collection and processing (52.2%).

D. RQ4: Gap Between the Current Research and Practitioners' Needs

After our literature review process, we identify a total of 38 papers. Table II shows the capabilities of state-of-the-art log anomaly detection techniques in terms of nine factors.

Data Resource. From Table II, most of these papers focus solely on log data for detecting anomalies. Specifically, 43 papers (91.5%) use historical labeled normal data, 23 papers (48.9%) use historical labeled abnormal data, and 3 papers [24]–[26] (7.9%) only rely on historical unlabeled data for training. In addition, 1 paper (2.1%) uses a few labeled normal and abnormal data as prompts for in-context learning with LLMs. However, our survey results reveal that only 39.3% and 43.5% of practitioners always have access to historical labeled normal and abnormal data, respectively. Thus, there is a noticeable gap between the availability of historical labeled data and its actual usage in real-world scenarios. Furthermore, only 4 papers (8.5%) integrate other types of data, such as metrics and traces, to aid in anomaly detection. Surprisingly, our survey shows that 83.7% and 74.9% of practitioners find metrics and traces always or sometimes available, respectively. Despite this availability, these data types are not utilized in the majority of studies. As one surveyed practitioner aptly stated: "*Different types of data are essential to train an effective anomaly log detection tool.*". This underscores the importance of considering these additional data types for anomaly detection.

Granularity level. Most of the papers evaluate their techniques on two public datasets, Hadoop Distributed File System (HDFS) [27] and Blue Gene/L supercomputer (BGL) [28], typically grouping logs into sequences using sessions and window sizes. Consequently, 34 (72.3%) and 30 (63.8%) studies operate at the log sequence level based on these grouping strategies, respectively. Additionally, 8 studies (17.0%) group logs according to time intervals. While 6 studies (12.8%) perform log event level anomaly detection, 3 of them [26], [29], [30] only work at the event level. Our survey results show that the top preference of practitioners surveyed (30.1%) is to work at the log sequence level based on window size, while 29.5% prefer to work at the log event level, which is the second preference.

> **Finding 9.** The majority (91%) of the studies rely on historical labeled normal log data for anomaly detection, which contrasts sharply with the surveyed data availability (39.3%). Few studies have incorporated other types of data for anomaly detection, which practitioners deem important. Additionally, a few studies work at the log event level, while 29.5% of surveyed practitioners prefer this granularity.

Effectiveness. The most important factor identified in the survey is the correct detection rates, measured by recall and precision. As demonstrated in Table II, 37 papers (78.7%) and 34 (72.3%) achieve recall and precision in the range

This article has been accepted for publication in IEEE Transactions on Software Engineering. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TSE.2025.3586700

of 80%-100%, respectively, meeting the requirements of all surveyed practitioners. 41 (87.2%) and 40 papers (85.1%) achieve recall and precision within 60%-80%, which can satisfy at least 60.9% and 57.7% of the surveyed practitioners, respectively. For the two papers [31], [32], the relatively low precision, which failed to reach 40%, may be attributed to the utilization of zero-shot and few-shot prompt learning, which often struggles with insufficient context. We categorize some papers as "?" in Table II since we cannot ascertain the detection rates of the log anomaly detection techniques they presented.

Efficiency. Our online survey (Figure 9) investigated practitioner expectations for both setup time (installation, configuration, and learning phases) and anomaly detection time. Regarding the installation, configuration, and learning time costs, our survey results reveal that an automated log anomaly detection technique with capabilities in the ranges of <10 minutes, 10-60 minutes, 1-3 hours, and 3-24 hours satisfies 100%, at least 73.1%, 40.4%, and 18.1% of the surveyed practitioners, respectively. Despite this clear preference, our literature review reveals that none of the surveyed papers report installation or configuration times, even among those offering log anomaly detection models in GitHub projects. This reporting gap likely stems from three key reasons: (1) inherent difficulties in standardizing measurements across diverse hardware/software environments, (2) academic incentives prioritizing novel detection methods over deployment practicality, and (3) the absence of established reporting standards for installation and configuration overhead. As one practitioner mentioned, "*If I find these tools and friendly install these tools, I will use these tools.*", emphasizing the need for greater attention to deployment practicality in research. In terms of the time required to detect anomalies, our survey results reveal that an automated log anomaly detection technique capable of identifying anomalies within 1 second meets the expectations of 100% of the surveyed practitioners, while a detection time of up to 5 seconds satisfies at least 61.2% of them. As illustrated in Table II, 14 papers (29.8%) fulfill the 1-second requirement, and 15 papers (31.9%) meet the 5-second criterion. Notably, 9 papers report detection times within 1 millisecond. However, more than half of the papers (68.1%) do not specify the testing time for their proposed techniques, resulting in their categorization as "?". This omission may arise from several factors: (1) the detection time may be so rapid that researchers overlook reporting it, or (2) there may be inconsistent measurement methodologies across different hardware/software environments.

Scalability. Our survey results point out that an automated log anomaly detection technique handling at least 1,000,000 logs satisfies 100% of the surveyed practitioners. Table II shows that 46 papers (i.e., excluding [26]) satisfy all surveyed practitioners, as they have evaluated their proposed techniques on at least one public dataset containing more than 1,000,000 logs.

TABLE II: The surveyed practitioners' capability expectations and the capabilities of current research.

Factor	Type		Papers
Data Resource	Historical normal log		[23], [33], [34], [31], [35], [36], [37], [5], [38], [39], [40], [41], [42], [43], [44], [3], [45], [46], [47], [48], [49], [50], [4], [6], [29], [51], [52], [53], [54], [55], [56], [57], [32], [58], [59], [60], [61], [62], [63], [64], [65], [66], [67]
	Historical abnormal log		[23], [33], [31], [5], [39], [40], [41], [42], [43], [3], [48], [49], [50], [54], [30], [32], [58], [59], [61], [62], [63], [65], [67]
	Historical unlabeled log		[24], [25], [26], [46], [30]
	Metric		[37], [5], [6]
	Trace		[37], [44]
Granularity	Event		[33], [50], [29], [25], [26], [30]
	Seq. (window size)		[23], [33], [34], [24], [36], [37], [39], [40], [41], [42], [43], [3], [45], [46], [48], [49], [4], [6], [25], [51], [54], [55], [56], [57], [32], [59], [61], [62], [63], [65]
	Seq. (time intervals)		[24], [5], [47], [52], [60], [64], [66], [67]
	Seq. (timestamp/session)		[23], [33], [34], [31], [24], [35], [36], [38], [39], [40], [41], [42], [43], [44], [3], [46], [49], [50], [4], [6], [51], [52], [53], [55], [56], [57], [58], [59], [60], [61], [64], [65], [66], [67]
Factor	**Sat. Rate[1]**		**Papers**
Effec.	Recall	80-100%	[23], [33], [34], [31], [24], [35], [36], [37], [5], [38], [39], [40], [41], [42], [43], [44], [3], [46], [47], [48], [49], [50], [4], [6], [25], [52], [54], [55], [56], [57], [32], [58], [60], [62], [63], [65], [66]
		60-80%	[51], [53], [61], [64]
		40-60%	[29]
		<40%	-
		?	[45], [30], [67], [59], [26]
	Precision	80-100%	[23], [33], [34], [24], [35], [37], [5], [38], [39], [40], [41], [43], [44], [3], [46], [47], [48], [49], [4], [6], [25], [52], [53], [55], [56], [57], [58], [60], [61], [62], [63], [64], [65], [66]
		60-80%	[36], [42], [50], [29], [51], [54]
		40-60%	-
		<40%	[31], [32]
		?	[45], [30], [67], [59], [26]
Effi.	<1 millisecond		[23], [31], [44], [46], [47], [6], [30], [58], [61]
	1 millisecond-1 second		[34], [24], [40], [25], [26]
	1-5 second		[37]
	>5 seconds		-
	?		[33], [35], [36], [5], [38], [39], [41], [42], [43], [3], [45], [48], [49], [50], [4], [29], [51], [52], [53], [54], [55], [56], [57], [32], [59], [60], [62], [63], [64], [65], [66], [67]
Scal.	≤1,000,000		[26]
	>1,000,000		[23], [33], [34], [31], [24], [35], [36], [37], [5], [38], [39], [40], [41], [42], [43], [44], [3], [45], [46], [47], [48], [49], [50], [4], [6], [29], [25], [51], [52], [53], [54], [55], [56], [30], [57], [32], [58], [59], [60], [61], [62], [63], [64], [65], [66], [67]
Factor	**Support**		**Papers**
Interpret.	Yes		[45], [32], [63], [64]
General.	Cross-project		[23], [33], [31], [48], [63]
Custom.	Yes		-
Priv.	Yes		-

[1] *Sat. Rate* represents the satisfaction rate of surveyed practitioners' choices. *Effec.*, *Effi.*, *Scal.*, *Interpret.*, *General.*, *Custom.*, and *Priv.* represent effectiveness, efficiency, scalability, interpretability, generalizability, customization, and privacy protection, respectively.

Interpretability. Most log anomaly detection techniques solely determine whether a testing log is anomalous or not. However, 84.5% surveyed practitioners desire more than just a binary classification; they want insights into the root causes of the detected anomalies (e.g., specific factors such as recent software updates, network congestion, or hardware failures), as this information would assist in system maintenance and enable them to take appropriate actions. Among the surveyed literature, only Zhao et al. [45] and Chai et al. [64] provide domain-knowledge reports, while Liu et al. [32] and Sui et al. [63] offer concise explanations for anomalies. Zhao et al. [45] proposed LogAD, which predicts anomalies while generating reports containing: time series of template counts, normal template sequences represented as finite state automatons (to understand task execution workflows), time series of specific variable values, and distributions of log variables using Jensen-Shannon divergence to measure differences between distributions. In their case study of illegal access detection in Nginx logs, LogAD identified the anomaly through both a spike in total log volume and distribution analysis, revealing that IP-1 and URL-a accounted for a large portion of requests. Similarly, Chai et al. [64] provided reports for detected anomalies that highlight the normal workflow most closely resembling the current abnormal workflow. Their reports visually present both the current abnormal workflow and the most likely normal workflow, offering engineers insights for diagnosing the anomalies. However, the reports primarily focus on domain knowledge and lack natural language explanations and human evaluation results, raising concerns about the actual effectiveness of the report's interpretability.

Liu's and Sui's works only provide a brief introduction to anomalies without offering the necessary underlying explanations. Liu et al. [32] employed LLMs with chain-of-thought learning to generate explanations for anomalies. For instance, one of their instructions states, "Mark it abnormal when and only when the alert is explicitly expressed in textual content (such as keywords like error or interrupt). Concisely explain your reason for each log.". While this approach identifies the presence of an anomaly, it does not elucidate the root causes or contributing factors, resulting in a lack of contextual details. Furthermore, the general nature of the instruction may lead to explanations that are often generic rather than providing the insightful analysis needed for effective understanding. Sui et al. [63] incorporated more detailed information in their reports, which include the original log sequence, interpretations, and relevant metadata such as timestamps and system identifiers. Their interpretations are generated using LLMs with in-context learning to highlight anomalies or irregularities that may indicate significant issues within the system, presented in natural language for ease of understanding. However, despite both works utilizing natural language to enhance clarity, they still fall short in providing deeper analysis, particularly regarding the root causes of the anomalies.

As a practitioner emphasized: "*The tool's ability to provide the rationale for a log being flagged as an anomaly is critical to understanding the underlying issue and taking appropriate action. Without this explanation, we have no way of knowing what happened, why, and how to fix it.*". Another practitioner noted, *"I truly need are actionable insights. For example, when an anomaly is flagged, I want to know which specific log lines played a role in that detection and the actionable guidance. This information helps me grasp the context and troubleshoot. Without this level of detail, I feel like I'm just guessing, especially as an engineer.*". Additionally, a practitioner expressed, *"I need detailed and clear explanations for the anomalies, like the likely reasons behind them and the specific conditions, such as whether they happen after some changes (e.g., system upgrades). Highlighting those details would really help me understand better.*". Nevertheless, none of the aforementioned papers adequately addresses these concerns or provides the necessary solutions to meet the practitioners' needs.

Generalizability. As training on a single dataset and testing on the same dataset typically result in a model learning a simplistic log structure, we explore studies to determine if they have conducted cross-project evaluations to demonstrate a degree of generalizability. According to our survey results, 83.8% of practitioners expect log anomaly detection techniques to possess generalizability, enabling them to effectively analyze logs with diverse structures or characteristics. As one practitioner noted, *"These tools won't be useful unless they can catch anomalies in a variety of situations."*. However, as indicated in Table II, only 5 papers [23], [31], [33], [48], [63] evaluate their techniques across different domains. These studies either train their models on a source domain and test them on a different target domain or utilize meta-learning strategies to adapt their models for various systems. While 2 papers [23], [48] achieve satisfactory results on public datasets like HDFS, these datasets have limited log information, addressing only one or two types of abnormal patterns [45]. For instance, BGL logs primarily contain information related to the RAS kernel, while HDFS logs describe operations on the storage block pool [31]. Moreover, publicly available data is typically single-sourced, and industrial datasets are often access-restricted due to security and privacy concerns, as highlighted by Lee et al. [5]. Hashemi et al. [33] also claim that achieving high accuracy is possible only when the target domain's data is sufficiently similar to the source domain's. Consequently, the lack of diverse log data and the limited scope of cross-project evaluations present significant challenges to achieving generalizability in log anomaly detection techniques. To further improve this area, collaboration between industry and researchers is essential to collect a broader range of log data that reflects various operational contexts and anomaly patterns. Additionally, researchers should focus on developing techniques that mitigate overfitting in DL-based models, ensuring that these models can generalize effectively across different datasets and scenarios.

Customization and privacy protection. None of the log anomaly detection techniques proposed in the 47 papers we reviewed offers customization and privacy protection. These two factors, which are crucial to most of the surveyed practitioners, have been overlooked in these studies. As some practitioners noted: "*Some exceptions are actively thrown out by the program, and some exceptions are real anomalies. How to determine whether they are actively thrown out for debug analysis is very important. That is, the tool should allow users* *to have different customization requirements.*”, “*Log data often includes user identifiers and IP addresses. Without proper privacy safeguards, our information could be exposed, so I would be concerned about this tool invading my privacy.*”.

> **Finding 10.** Most of the automated log anomaly detection techniques proposed in the surveyed papers achieve recall and precision within 80%-100%, meeting the expectations of all practitioners. 93% of the techniques that report detection times are able to detect anomalies within one second, satisfying all practitioners' expectations for anomaly detection time. However, few or no papers address practitioners' needs for interpretability, generalizability, customization, and privacy protection.

Challenges of adopting existing log anomaly detection techniques. Although most of the state-of-the-art log anomaly detection techniques reviewed in our study achieve high accurate detection rates (i.e., recall and precision within 80%-100%), around half of the surveyed practitioners (51%) do not prioritize these techniques due to various concerns expressed at the end of the survey. We categorize three main types of concerns and present their frequencies using the multiplication symbol (*X*). ***(a) Lack of interpretability (26X)*:** Many practitioners note that existing techniques often provide only binary results without explanations, which they find insufficient. “*I will consider using these models if they provide hints for me to debug my system.*” and “*The models in research are not a priority because the unexplainability introduced by deep learning algorithms cannot explain its predictions.*”. ***(b) Unsure capability of handling various log data (19X)*:** Some practitioners point out that existing techniques are trained on limited datasets that may not represent industrial data adequately. “*I will not use these models. The currently used datasets are still relatively one-sided and cannot widely represent the distribution of data sets in the industry.*”. ***(c) Lack of user-friendly usage (12X)*:** Several surveyed practitioners express concerns about the complexity of techniques and whether they can be used easily. “*The existing log anomaly detection techniques often require expertise in model tuning, making them too challenging for me to try.*”.

IV. Discussion

A. Implications

Our results highlight some key implications for research communities:

(1) Large improvement in interpretability and generalizability of existing techniques is needed. To achieve at least an 80% satisfaction rate, these techniques for automated log anomaly detection need to provide a rationale for their detection and efficiently detect various log anomalies. However, our literature review reveals that the majority of the surveyed techniques overlook these factors. Notably, only two papers offer explanations for detected anomalies. While some practitioners emphasize the importance of generalizability for adopting these techniques, only four papers evaluate the generalizability of their techniques across different projects. However, the effectiveness of these techniques remains unclear or unsatisfactory due to limited datasets and single log structure. Thus, we encourage researchers to consider providing interpretable results for detected anomalies and developing techniques capable of handling diverse log data.

(2) Community-wide effort to integrate state-of-the-art techniques into automated log anomaly detection tools with customization is needed. Through our survey, the majority of practitioners (78.2%) prioritize customization in automated log anomaly detection tools, such as adapting anomaly alert thresholds, as detailed in Section III-C. Nevertheless, none of the techniques proposed in the surveyed papers discuss this issue. To bridge this gap, a collective effort from the community is essential to develop automated tools that offer robust customization options that better meet industry needs. Such an effort should involve practitioners clearly articulating their domain-specific requirements (e.g., configurable thresholds), while researchers focus on designing adaptable frameworks that accommodate these needs through modular and extensible architectures. “*We can consider using the automated tools. However, given the complexity of our business, we are uncertain about their ability to adapt to our specific task of identifying real anomalies, such as an alert threshold.*”.

(3) Suggestions for LLM-based automated log anomaly detection. We have encouraged practitioners to share their insights on the potential of LLMs to aid in the development of log anomaly detection techniques, given their increasing popularity for software engineering tasks [68]–[70], particularly in log analysis-related activities, such as employing zero-shot learning for log parsing [71] and integrating interpretable domain knowledge through continual pre-training to enhance LLM capabilities for log analysis [72]. Recent studies (e.g., [32], [59], [60], [63]–[66]) demonstrate that LLMs improve anomaly detection through self-supervised pre-training on large-scale datasets and strong learning capabilities. However, key challenges remain, as highlighted by practitioner feedback, which we summarize for further investigation: (a) Integrating LLMs as a plug-in function to improve interpretability: “*It should plug in with my existing tool instead of being an add-on tool to provide an explanation.*” (b) Utilizing LLMs' in-context learning with high-quality data: “*The accuracy of your tool to detect anomalies in the industry is very, very important. Even one false anomaly detection may have serious consequences. Additionally, I'm worried about whether the training time costs will end up being passed on to us users. While prompt learning could avoid the need for fine-tuning, the data used for prompting needs to be carefully chosen because it has a significant impact on LLM performance.*” (c) Addressing data security challenges for in-context learning: “*My main concern about using LLMs is data security. If I use in-context learning to prompt LLMs, will the inference process of LLMs expose my data? Are there sufficient safeguards in place to protect my data?*”. In conclusion, LLMs can function as integrated enhancements to existing tools, enabling practitioners to maintain their current systems, which mainly benefit from the advanced capabilities of LLMs. It is also crucial to ensure that the data used is both high-quality and secure. Given that the data may have quality issues [73], researchers should focus on selecting high-quality data for building effective log anomaly detection models. Additionally, to ensure data security, researchers need to develop privacy protection methods that minimize the risk of data leakage while maintaining data utility.

(4) Potential methods for further improvement. Regarding ***interpretability***, integrating post-hoc eXplainable Artificial Intelligence (XAI) methods, such as Local Interpretable Model-agnostic Explanations (LIME) [74], SHapley Additive exPlanations (SHAP) [75], and Anchors [76], with DL models can enhance the understanding of model behavior. These techniques aim to provide feature-level explanations by approximating the behavior of the original model with an interpretable model [77], [78], which are also valuable for log anomaly detection. By seamlessly incorporating these XAI methods into existing DL-based tools, we suggest enhancing model interpretability by identifying critical log components (e.g., tokens) that drive log anomaly detection decisions. Furthermore, generative AI models, such as ChatGPT [79] and DeepSeek [80], demonstrate exceptional performance and present an opportunity to develop an XAI framework that enhances interpretability. Once current models or tools identify anomalies, these anomalies, along with their most influential tokens identified through XAI methods, can be processed within this framework for further analysis. To facilitate this process, the framework can curate a subset of specific log anomalies, each accompanied by human-written explanations. Given the impracticality of collecting all possible anomalies, techniques such as clustering-based algorithms can be employed to sample a more representative and manageable set of log anomalies, for which domain experts will provide detailed explanations. For the test log data, distance calculations (such as k-Nearest Neighbors and Euclidean distance) are performed to select the top-k log instances from the curated subset of log anomalies, as few-shot examples. The framework can then leverage these few-shot examples and the few-shot learning capabilities of LLMs to better analyze and determine the causes of anomalies in the test log data, thereby enhancing the interpretability of log anomaly detection systems. In terms of ***customization***, existing tools typically use a fixed alert threshold for anomaly detection. We recommend incorporating an adjustable option that empowers users to modify the sensitivity of the anomaly detection process. This flexibility would allow users to increase or decrease the threshold based on their assessment of alert frequency, helping to reduce alert fatigue or ensure that critical anomalies are not overlooked. Additionally, we suggest the integration of domain-specific rules tailored for various operational needs. This could include the identification of critical keywords that require heightened attention from the log anomaly detection tools, as well as the specification of time windows for log sequence analysis.

Our results highlight key implications for industrial communities:

(1) Large demand for user-friendly automated tools is needed. Ease of use is a major concern among surveyed practitioners, with at least 76.6% considering it important (Section III-C). This includes aspects such as the tool's installation and deployment, with only 18% willing to invest more than three hours in these tasks. Therefore, there is a clear need for the development of easy-to-use automated tools that incorporate advanced techniques while also providing comprehensive user guides and well-designed interfaces. "*If the automated log anomaly detection tool lacks a user interface, I may not consider it, as convenience is key for me. I prefer tools that are readily available and easy to use.*". Moreover, over half of the surveyed practitioners stress compatibility issues when using log monitoring tools for log anomaly detection. "*The automated tool should feature a modular architecture to ensure easy integration into our existing systems. Alternatively, providing APIs to facilitate integration with diverse legacy systems would greatly benefit us, enabling smooth data exchange and enhancing overall interoperability.*". Consequently, prioritizing the development of user-friendly and compatible automated log anomaly detection tools is essential.

(2) Data quality improvement is required for further technique design. At the conclusion of our survey, we invite any additional considerations regarding log anomaly detection or automated log anomaly detection tools. Several practitioners have raised concerns about the quality of log data utilized in current research, indicating that it may not adequately capture exceptions or represent the broader context. As one practitioner remarked, "*First of all, the log information is still relatively simple, and some exceptions are difficult to detect from the log content (such as delays caused by excessive load). Secondly, the currently used data sets for training models are still relatively one-sided, and can not widely represent the distribution of data sets in the industry, and there are still certain limitations.*". Another practitioner emphasized, "*There is no usage of these public datasets used in academia.*". These comments highlight significant apprehensions about the data employed in developing log anomaly detection models. Moreover, several academic papers have underscored this data quality issue. Landauer et al. [73] conducted an in-depth examination of this concern, arguing that commonly used public datasets (such as HDFS [27] and BGL [28]) or collections of log datasets (e.g., Loghub [81]) for log analytics in research are often simplistic, allowing straightforward techniques to achieve high detection rates. The study reveals that these datasets often lack complexity and diversity, and any anomalies present in these datasets are relatively straightforward to detect, often affecting only specific event types and failing to account for the sequence of events. This oversimplification may lead to a narrow understanding of anomaly detection, as it does not reflect the multifaceted nature of anomalies encountered in operational environments. To address these challenges, we recommend that practitioners preprocess their collected log data (e.g., removing sensitive information) and subsequently publish the log datasets. This initiative would provide researchers with access to more diverse and representative industrial log data, enabling them to develop and propose techniques that lead to more robust and generalizable models in log anomaly detection.

B. Follow-up Survey

To validate our proposed implications, we conducted a follow-up survey with the same professionals who participated in our initial interviews. This approach ensures continuity in our research and allows us to track evolving perspectives on log anomaly detection.

(1) In terms of interpretability, an overwhelming 93.3% of professionals agree that explainable detection results are essential for adoption. Many note that current explanations tend to be overly technical or lack actionable insights. Regarding generalizability, 80.0% concur with our implication that there is a need for more diverse training data, with some expressing that most academic solutions struggle when applied to their heterogeneous log systems. (2) Our recommendation for customizable tools receives unanimous support (100%). Professionals particularly value the ability to adjust thresholds (86.7%), while 60.0% emphasize the necessity for deeper customization, such as integrating business rules. Many express frustration with the rigidity of existing tools, highlighting that customization is vital for practical applications. (3) While 66.7% acknowledge the potential of LLMs for improved anomaly explanation, adoption remains low due to security concerns, particularly among those hesitant to use cloud-based LLMs, and operational costs. Several professionals suggest hybrid approaches where LLMs process only preprocessed log data (e.g., data with sensitive information removed). (4) The demand for user-friendly tools is pronounced, with 100% prioritizing easy deployment and intuitive interfaces. Regarding data quality, 93.3% agree with our suggestion that industrial practitioners could provide more representative log data for research.

In summary, our interviewed professionals confirmed that our implications accurately identify key challenges in log anomaly detection. Their feedback underscores specific pain points related to explanation clarity, customization depth, and real-world applicability, which should inform future research directions.

C. Threats to Validity

One of the potential threats to the validity of our survey is the possibility that some practitioners may not fully comprehend the questions. For example, some practitioners may not understand log anomaly detection and instead rely solely on automated tools to manage logs. Therefore, they may be unfamiliar with the questions about log anomaly detection. To mitigate this threat, we provided high-level instructions accompanied by a flow chart to clarify the questions. Furthermore, responses indicating "I don't know" will be excluded from the analysis. This threat to validity is common and considered tolerable in previous studies [82], [83]. Another potential threat to the validity of our study is that our sample of practitioners does not encompass the entire population of software engineers. Specifically, our practitioners are limited to those working for various companies and contributors to open-source projects hosted on GitHub in diverse roles. Consequently, our findings may not fully represent the expectations of all software engineers. For instance, our survey excludes practitioners who are not proficient in either English or Chinese. While we focus on several factors that may influence the adoption of automated log anomaly detection tools, there may be additional factors that our study has not addressed. We plan to investigate these factors in future research.

V. Related Work

A. Automated Log Anomaly Detection

Numerous ML and DL approaches have been proposed for automated log anomaly detection, including supervised, semi-supervised, and unsupervised approaches. Supervised approaches, leveraging abundant labeled data, typically outperform semi-supervised and unsupervised methods. Liang et al. [84] introduced four classifiers for predicting failure log events. Zhang et al. [85] represented log templates as vectors combined with log template extraction with $tf-idf$, employing an LSTM model for anomaly prediction. Similarly, Vinayakumar et al. [86] utilized a stacked-LSTM model to learn temporal patterns with sparse representations. Lu et al. [87] learned local semantic information from log data utilizing CNN with three filters. Zhang et al. [50] integrated attention mechanisms with a Bi-LSTM model to capture comprehensive log sequences bidirectionally. Le et al. [3] leveraged BERT for log representation and a Transformer encoder for log anomaly detection. Semi-supervised and unsupervised approaches, requiring less labeled data, are considered more practical. For instance, Du et al. [6] autonomously learned log patterns with an LSTM model and detected anomalies based on deviations from normal execution. Meng et al. [4] incorporated semantic information by matching log sequences against generated templates to learn normal patterns. Yang et al. [46] combined HDBSCAN clustering for probabilistic label estimation with an attention-based GRU model. Huo et al. [40] examined logging statements and constructed execution graphs to identify log-related paths. Anomaly labels are then propagated to each execution path based on the analysis of labeled sequence anomalies. Lin et al. [88] proposed an unsupervised approach that clusters log sequences hierarchically, taking into account the weights of log events. Lee et al. [89] learned log information autonomously without parsing based on BERT. Yang et al. [24] incorporated a lightweight semantic-based log representation in traditional unsupervised principal component analysis for log anomaly detection. Liu et al. [90] introduced a scalable and adaptive log-based anomaly detection framework for cloud systems that utilizes a trie-based detection agent for efficient streaming detection and integrates LLMs (e.g., ChatGPT) as expert feedback to improve accuracy.

B. Studies on Log Analysis

Several recent studies have investigated log analysis through empirical methods. For instance, Fu et al. [91] surveyed 54 experienced developers at Microsoft to investigate logging practices, particularly where developers choose to log. Chen et al. [92] reviewed and evaluated five popular neural networks used by six DL-based log anomaly detection models. He et al. [93] examined the four main steps in the automated log analysis framework: logging, log compression, log parsing, and log mining. Li et al. [94] employed semi-structured interviews and surveys to examine practitioners' expectations regarding the readability of log messages and to explore the potential for automatically classifying the readability of these messages. Yang et al. [95] conducted interviews to understand how developers utilize logs within an embedded software engineering context. Rong et al. [96] carried out an empirical study using mixed methods, including questionnaire surveys, semi-structured interviews, and code analyses, to explore the relationships between developers' profiles, experiences, and their logging practices. However, there is a lack of research focused on the practices, issues, and expectations of practitioners regarding automated log anomaly detection tools. In contrast, our work examines practitioners' expectations for log anomaly detection tools from various aspects (e.g., granularity, detection accuracy, real-time capability) and explores thresholds for effectiveness, efficiency, and scalability. In addition, we also conduct a comprehensive literature review to identify gaps between current techniques and practitioners' expectations.

VI. Conclusion

In this paper, we interview 15 professionals and survey 312 practitioners about their log anomaly detection practices, the issues they face, and their expectations for automated log anomaly detection tools. Practitioners express dissatisfaction with existing log monitoring tools for log anomaly detection and indicate a willingness to adopt automated log anomaly detection tools if certain aspects are satisfied, including granularity level, evaluation metrics, effectiveness, efficiency, scalability, and privacy protection. We also compare the capabilities of current research with practitioners' expectations through a literature review, offering insights for further improvements to more effectively meet practitioners' needs.

Acknowledgement

We extend our gratitude to the participants who took part in our study, providing valuable responses to our survey questions and offering insightful feedback. This work obtained ethics approval by the Research Ethics Committee of the Hong Kong Metropolitan University (HE-SF2025/19).

References

[1] D. M. German, P. C. Rigby, and M.-A. Storey, "Using evolutionary annotations from change logs to enhance program comprehension," in *Proceedings of the 2006 international workshop on Mining software repositories*, 2006, pp. 159–162.

[2] Z. Ding, Y. Tang, Y. Li, H. Li, and W. Shang, "On the temporal relations between logging and code," in *2023 IEEE/ACM 45th International Conference on Software Engineering (ICSE)*. IEEE, 2023, pp. 843–854.

[3] Le, Van-Hoang and Zhang, Hongyu, "Log-based anomaly detection without log parsing," in *2021 36th IEEE/ACM International Conference on Automated Software Engineering (ASE)*. IEEE, 2021, pp. 492–504.

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

[5] C. Lee, T. Yang, Z. Chen, Y. Su, Y. Yang, and M. R. Lyu, "Heterogeneous anomaly detection for software systems via semi-supervised cross-modal attention," in *2023 IEEE/ACM 45th International Conference on Software Engineering (ICSE)*. IEEE, 2023, pp. 1724–1736.

[6] M. Du, F. Li, G. Zheng, and V. Srikumar, "Deeplog: Anomaly detection and diagnosis from system logs through deep learning," in *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security (CCS)*, 2017, pp. 1285–1298.

[7] A. R. Chen, T.-H. Chen, and S. Wang, "Pathidea: Improving information retrieval-based bug localization by re-constructing execution paths using logs," *IEEE Transactions on Software Engineering*, vol. 48, no. 8, pp. 2905–2919, 2021.

[8] X. Zhou, X. Peng, T. Xie, J. Sun, C. Ji, D. Liu, Q. Xiang, and C. He, "Latent error prediction and fault localization for microservice applications by learning from system trace logs," in *Proceedings of the 2019 27th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering*, 2019, pp. 683–694.

[9] B. Yu, J. Yao, Q. Fu, Z. Zhong, H. Xie, Y. Wu, Y. Ma, and P. He, "Deep learning or classical machine learning? an empirical study on log-based anomaly detection," in *Proceedings of the 46th IEEE/ACM International Conference on Software Engineering*, 2024, pp. 1–13.

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

[11] X. Ma, H. Zou, P. He, J. Keung, Y. Li, X. Yu, and F. Sarro, "On the influence of data resampling for deep learning-based log anomaly detection: Insights and recommendations," *IEEE Transactions on Software Engineering*, 2024.

[12] P. Leavy, *Research design: Quantitative, qualitative, mixed methods, arts-based, and community-based participatory research approaches*. Guilford Publications, 2022.

[13] C. B. Seaman, "Qualitative methods in empirical studies of software engineering," *IEEE Transactions on software engineering*, vol. 25, no. 4, pp. 557–572, 1999.

[14] S. Baltes and P. Ralph, "Sampling in software engineering research: A critical review and guidelines," *Empirical Software Engineering*, vol. 27, no. 4, p. 94, 2022.

[15] D. Spencer, *Card sorting: Designing usable categories*. Rosenfeld Media, 2009.

[16] "Google forms," https://docs.google.com/forms, 2024.

[17] "Survey form," https://figshare.com/articles/online_resource/Survey/25981564, 2024, [Online].

[18] "Wenjuanxing software," https://www.wjx.cn, 2024.

[19] M. Fahmideh, A. Ahmad, A. Behnaz, J. Grundy, and W. Susilo, "Software engineering for internet of things: The practitioners' perspective," *IEEE Transactions on Software Engineering*, vol. 48, no. 8, pp. 2857–2878, 2021.

[20] M. Hort, M. Kechagia, F. Sarro, and M. Harman, "A survey of performance optimization for mobile applications," *IEEE Transactions on Software Engineering*, vol. 48, no. 8, pp. 2879–2904, 2021.

[21] R. Moussa and F. Sarro, "On the use of evaluation measures for defect prediction studies," in *Proceedings of the 31st ACM SIGSOFT International Symposium on Software Testing and Analysis*, 2022, pp. 101–113.

[22] C. Wohlin, "Guidelines for snowballing in systematic literature studies and a replication in software engineering," in *Proceedings of the 18th international conference on evaluation and assessment in software engineering*, 2014, pp. 1–10.

[23] H. Guo, J. Yang, J. Liu, J. Bai, B. Wang, Z. Li, T. Zheng, B. Zhang, J. Peng, and Q. Tian, "Logformer: A pre-train and tuning pipeline for log anomaly detection," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 38, no. 1, 2024, pp. 135–143.

[24] L. Yang, J. Chen, S. Gao, Z. Gong, H. Zhang, Y. Kang, and H. Li, "Try with simpler–an evaluation of improved principal component analysis in log-based anomaly detection," *ACM Transactions on Software Engineering and Methodology*, 2023.

[25] A. Farzad and T. A. Gulliver, "Unsupervised log message anomaly detection," *ICT Express*, vol. 6, no. 3, pp. 229–237, 2020.

[26] J. Kim, V. Savchenko, K. Shin, K. Sorokin, H. Jeon, G. Pankratenko, S. Markov, and C.-J. Kim, "Automatic abnormal log detection by analyzing log history for providing debugging insight," in *Proceedings of the ACM/IEEE 42nd International Conference on Software Engineering: Software Engineering in Practice*, 2020, pp. 71–80.

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

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

[29] S. Nedelkoski, J. Bogatinovski, A. Acker, J. Cardoso, and O. Kao, "Self-attentive classification-based anomaly detection in unstructured logs," in

2020 IEEE International Conference on Data Mining (ICDM). IEEE, 2020, pp. 1196–1201.

[30] W. Meng, Y. Liu, S. Zhang, F. Zaiter, Y. Zhang, Y. Huang, Z. Yu, Y. Zhang, L. Song, M. Zhang *et al.*, "Logclass: Anomalous log identification and classification with partial labels," *IEEE Transactions on Network and Service Management*, vol. 18, no. 2, pp. 1870–1884, 2021.

[31] C. Zhang, T. Jia, G. Shen, P. Zhu, and Y. Li, "Metalog: Generalizable cross-system anomaly detection from logs with meta-learning," in *Proceedings of the IEEE/ACM 46th International Conference on Software Engineering*, 2024, pp. 1–12.

[32] Y. Liu, S. Tao, W. Meng, J. Wang, W. Ma, Y. Chen, Y. Zhao, H. Yang, and Y. Jiang, "Interpretable online log analysis using large language models with prompt strategies," in *Proceedings of the 32nd IEEE/ACM International Conference on Program Comprehension*, 2024, pp. 35–46.

[33] S. Hashemi and M. Mäntylä, "Onelog: towards end-to-end software log anomaly detection," *Automated Software Engineering*, vol. 31, no. 2, p. 37, 2024.

[34] Y. Xie, H. Zhang, and M. A. Babar, "Logsd: Detecting anomalies from system logs through self-supervised learning and frequency-based masking," *arXiv preprint arXiv:2404.11294*, 2024.

[35] 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 Software Engineering*, vol. 30, no. 1, p. 4, 2023.

[36] 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 *2023 38th IEEE/ACM International Conference on Automated Software Engineering (ASE)*. IEEE, 2023, pp. 141–152.

[37] J. Huang, Y. Yang, H. Yu, J. Li, and X. Zheng, "Twin graph-based anomaly detection via attentive multi-modal learning for microservice system," in *2023 38th IEEE/ACM International Conference on Automated Software Engineering (ASE)*. IEEE, 2023, pp. 66–78.

[38] 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 Transactions on Services Computing*, 2023.

[39] Y. Fu, K. Liang, and J. Xu, "Mlog: Mogrifier lstm-based log anomaly detection approach using semantic representation," *IEEE Transactions on Services Computing*, 2023.

[40] Y. Huo, Y. Li, Y. Su, P. He, Z. Xie, and M. R. Lyu, "Autolog: A log sequence synthesis framework for anomaly detection," in *2023 38th IEEE/ACM International Conference on Automated Software Engineering (ASE)*. IEEE, 2023, pp. 497–509.

[41] W. Wang, S. Lu, J. Luo, and C. Wu, "Deepuserlog: Deep anomaly detection on user log using semantic analysis and key-value data," in *2023 IEEE 34th International Symposium on Software Reliability Engineering (ISSRE)*. IEEE, 2023, pp. 172–182.

[42] 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 *2023 IEEE 34th International Symposium on Software Reliability Engineering (ISSRE)*. IEEE, 2023, pp. 194–206.

[43] S. Hashemi and M. Mäntylä, "Sialog: detecting anomalies in software execution logs using the siamese network," *Automated Software Engineering*, vol. 29, no. 2, p. 61, 2022.

[44] C. Zhang, X. Peng, C. Sha, K. Zhang, Z. Fu, X. Wu, Q. Lin, and D. Zhang, "Deeptralog: Trace-log combined microservice anomaly detection through graph-based deep learning," in *Proceedings of the 44th International Conference on Software Engineering*, 2022, pp. 623–634.

[45] N. Zhao, H. Wang, Z. Li, X. Peng, G. Wang, Z. Pan, Y. Wu, Z. Feng, X. Wen, W. Zhang *et al.*, "An empirical investigation of practical log anomaly detection for online service systems," in *Proceedings of the 29th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering*, 2021, pp. 1404–1415.

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

[47] 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 *2021 IEEE 32nd International Symposium on Software Reliability Engineering (ISSRE)*. IEEE, 2021, pp. 80–90.

[48] 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 *2020 IEEE 31st International Symposium on Software Reliability Engineering (ISSRE)*. IEEE, 2020, pp. 37–47.

[49] 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 *2020 IEEE 31st International Symposium on Software Reliability Engineering (ISSRE)*. IEEE, 2020, pp. 92–103.

[50] X. Zhang, Y. Xu, Q. Lin, B. Qiao, H. Zhang, Y. Dang, C. Xie, X. Yang, Q. Cheng, Z. Li *et al.*, "Robust log-based anomaly detection on unstable log data," in *Proceedings of the 2019 27th ACM joint meeting on European software engineering conference and symposium on the foundations of software engineering*, 2019, pp. 807–817.

[51] 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 *Proceedings of the 27th ACM SIGKDD conference on knowledge discovery & data mining*, 2021, pp. 3726–3734.

[52] S. Zhang, Y. Liu, X. Zhang, W. Cheng, H. Chen, and H. Xiong, "Cat: beyond efficient transformer for content-aware anomaly detection in event sequences," in *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 2022, pp. 4541–4550.

[53] T. Jia, P. Chen, L. Yang, Y. Li, F. Meng, and J. Xu, "An approach for anomaly diagnosis based on hybrid graph model with logs for distributed services," in *2017 IEEE international conference on web services (ICWS)*. IEEE, 2017, pp. 25–32.

[54] C. Duan, T. Jia, Y. Li, and G. Huang, "Aclog: An approach to detecting anomalies from system logs with active learning," in *2023 IEEE International Conference on Web Services (ICWS)*. IEEE, 2023, pp. 436–443.

[55] 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 *2020 IEEE International Conference on Software Maintenance and Evolution (ICSME)*. IEEE, 2020, pp. 667–671.

[56] 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 *Proceedings of the 38th Annual Computer Security Applications Conference*, 2022, pp. 769–784.

[57] 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 Transactions on Network and Service Management*, vol. 18, no. 4, pp. 4119–4133, 2021.

[58] G. Chu, J. Wang, Q. Qi, H. Sun, Z. Zhuang, B. He, Y. Jing, L. Zhang, and J. Liao, "Anomaly detection on interleaved log data with semantic association mining on log-entity graph," *IEEE Transactions on Software Engineering*, 2025.

[59] Y. Liu, Y. Ji, S. Tao, M. He, W. Meng, S. Zhang, Y. Sun, Y. Xie, B. Chen, and H. Yang, "Loglm: From task-based to instruction-based automated log analysis," *2025 IEEE/ACM 47th International Conference on Software Engineering (ICSE)*, 2025.

[60] C. Almodovar, F. Sabrina, S. Karimi, and S. Azad, "Logfit: Log anomaly detection using fine-tuned language models," *IEEE Transactions on Network and Service Management*, vol. 21, no. 2, pp. 1715–1723, 2024.

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

[62] M. He, T. Jia, C. Duan, H. Cai, Y. Li, and G. Huang, "Llmelog: An approach for anomaly detection based on llm-enriched log events," in *2024 IEEE 35th International Symposium on Software Reliability Engineering (ISSRE)*. IEEE, 2024, pp. 132–143.

[63] Y. Sui, X. Wang, T. Cui, T. Xiao, C. He, S. Zhang, Y. Zhang, X. Yang, Y. Sun, and D. Pei, "Bridging the gap: Llm-powered transfer learning for log anomaly detection in new software systems," in *2025 IEEE 41st International Conference on Data Engineering (ICDE)*. IEEE, 2025.

[64] X. Chai, H. Zhang, J. Zhang, Y. Sun, and S. K. Das, "Log sequence anomaly detection based on template and parameter parsing via bert," *IEEE Transactions on Dependable and Secure Computing*, 2024.

[65] 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 *2023 IEEE 34th International Symposium on Software Reliability Engineering (ISSRE)*. IEEE, 2023, pp. 207–217.

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

[67] S. Andonov and G. Madjarov, "Loggc: Novel approach for graph-based log anomaly detection," in *2023 IEEE International Conference on Data Mining Workshops (ICDMW)*. IEEE, 2023, pp. 1194–1202.

[68] J. Wang, Y. Huang, C. Chen, Z. Liu, S. Wang, and Q. Wang, "Software testing with large language models: Survey, landscape, and vision," *IEEE Transactions on Software Engineering*, 2024.

[69] X. Du, M. Liu, K. Wang, H. Wang, J. Liu, Y. Chen, J. Feng, C. Sha, X. Peng, and Y. Lou, "Evaluating large language models in class-level code generation," in *Proceedings of the IEEE/ACM 46th International Conference on Software Engineering*, 2024, pp. 1–13.

[70] A. Fan, B. Gokkaya, M. Harman, M. Lyubarskiy, S. Sengupta, S. Yoo, and J. M. Zhang, "Large language models for software engineering: Survey and open problems," *arXiv preprint arXiv:2310.03533*, 2023.

[71] X. Yu, S. Nong, D. He, W. Zheng, T. Ma, N. Liu, J. Li, and G. Xie, "Loggenius: An unsupervised log parsing framework with zero-shot prompt engineering," in *2024 IEEE International Conference on Web Services (ICWS)*. IEEE, 2024, pp. 1321–1328.

[72] Y. Ji, Y. Liu, F. Yao, M. He, S. Tao, X. Zhao, S. Chang, X. Yang, W. Meng, Y. Xie *et al.*, "Adapting large language models to log analysis with interpretable domain knowledge," *arXiv preprint arXiv:2412.01377*, 2024.

[73] M. Landauer, F. Skopik, and M. Wurzenberger, "A critical review of common log data sets used for evaluation of sequence-based anomaly detection techniques," *Proceedings of the ACM on Software Engineering*, vol. 1, no. FSE, pp. 1354–1375, 2024.

[74] M. T. Ribeiro, S. Singh, and C. Guestrin, "" why should i trust you?" explaining the predictions of any classifier," in *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, 2016, pp. 1135–1144.

[75] S. M. Lundberg and S.-I. Lee, "A unified approach to interpreting model predictions," *Advances in neural information processing systems*, vol. 30, 2017.

[76] M. T. Ribeiro, S. Singh, and C. Guestrin, "Anchors: High-precision model-agnostic explanations," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 32, no. 1, 2018.

[77] A. M. Antoniadi, Y. Du, Y. Guendouz, L. Wei, C. Mazo, B. A. Becker, and C. Mooney, "Current challenges and future opportunities for xai in machine learning-based clinical decision support systems: a systematic review," *Applied Sciences*, vol. 11, no. 11, p. 5088, 2021.

[78] L. Longo, M. Brcic, F. Cabitza, J. Choi, R. Confalonieri, J. Del Ser, R. Guidotti, Y. Hayashi, F. Herrera, A. Holzinger *et al.*, "Explainable artificial intelligence (xai) 2.0: A manifesto of open challenges and interdisciplinary research directions," *Information Fusion*, vol. 106, p. 102301, 2024.

[79] J. Achiam, S. Adler, S. Agarwal, L. Ahmad, I. Akkaya, F. L. Aleman, D. Almeida, J. Altenschmidt, S. Altman, S. Anadkat *et al.*, "Gpt-4 technical report," *arXiv preprint arXiv:2303.08774*, 2023.

[80] A. Liu, B. Feng, B. Xue, B. Wang, B. Wu, C. Lu, C. Zhao, C. Deng, C. Zhang, C. Ruan *et al.*, "Deepseek-v3 technical report," *arXiv preprint arXiv:2412.19437*, 2024.

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

[82] C. Wang, J. Hu, C. Gao, Y. Jin, T. Xie, H. Huang, Z. Lei, and Y. Deng, "Practitioners' expectations on code completion," *arXiv preprint arXiv:2301.03846*, 2023.

[83] Y. Yang, Y. Jiang, M. Gu, J. Sun, J. Gao, and H. Liu, "A language model for statements of software code," in *2017 32nd IEEE/ACM International Conference on Automated Software Engineering (ASE)*. IEEE, 2017, pp. 682–687.

[84] Y. Liang, Y. Zhang, H. Xiong, and R. Sahoo, "Failure prediction in ibm bluegene/l event logs," in *7th IEEE International Conference on Data Mining (ICDM)*. IEEE, 2007, pp. 583–588.

[85] K. Zhang, J. Xu, M. R. Min, G. Jiang, K. Pelechrinis, and H. Zhang, "Automated it system failure prediction: A deep learning approach," in *2016 IEEE International Conference on Big Data (Big Data)*. IEEE, 2016, pp. 1291–1300.

[86] R. Vinayakumar, K. Soman, and P. Poornachandran, "Long short-term memory based operation log anomaly detection," in *2017 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*. IEEE, 2017, pp. 236–242.

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

[88] Q. Lin, H. Zhang, J.-G. Lou, Y. Zhang, and X. Chen, "Log clustering based problem identification for online service systems," in *Proceedings of the 38th International Conference on Software Engineering Companion*, 2016, pp. 102–111.

[89] Y. Lee, J. Kim, and P. Kang, "Lanobert: System log anomaly detection based on bert masked language model," *Applied Soft Computing*, vol. 146, p. 110689, 2023.

[90] J. Liu, J. Huang, Y. Huo, Z. Jiang, J. Gu, Z. Chen, C. Feng, M. Yan, and M. R. Lyu, "Log-based anomaly detection based on evt theory with feedback," *arXiv preprint arXiv:2306.05032*, 2023.

[91] Q. Fu, J. Zhu, W. Hu, J.-G. Lou, R. Ding, Q. Lin, D. Zhang, and T. Xie, "Where do developers log? an empirical study on logging practices in industry," in *Companion Proceedings of the 36th International Conference on Software Engineering*, 2014, pp. 24–33.

[92] Z. Chen, J. Liu, W. Gu, Y. Su, and M. R. Lyu, "Experience report: Deep learning-based system log analysis for anomaly detection," *arXiv preprint arXiv:2107.05908*, 2021.

[93] S. He, P. He, Z. Chen, T. Yang, Y. Su, and M. R. Lyu, "A survey on automated log analysis for reliability engineering," *ACM computing surveys (CSUR)*, vol. 54, no. 6, pp. 1–37, 2021.

[94] Z. Li, A. R. Chen, X. Hu, X. Xia, T.-H. Chen, and W. Shang, "Are they all good? studying practitioners' expectations on the readability of log messages," in *2023 38th IEEE/ACM International Conference on Automated Software Engineering (ASE)*. IEEE, 2023, pp. 129–140.

[95] N. Yang, P. Cuijpers, D. Hendriks, R. Schiffelers, J. Lukkien, and A. Serebrenik, "An interview study about the use of logs in embedded software engineering," *Empirical Software Engineering*, vol. 28, no. 2, p. 43, 2023.

[96] G. Rong, S. Gu, H. Shen, H. Zhang, and H. Kuang, "How do developers' profiles and experiences influence their logging practices? an empirical study of industrial practitioners," in *2023 IEEE/ACM 45th International Conference on Software Engineering (ICSE)*. IEEE, 2023, pp. 855–867.