IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received January 10, 2022, accepted February 9, 2022, date of publication February 17, 2022, date of current version March 3, 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3152549

Landscape of Automated Log Analysis: A Systematic Literature Review and Mapping Study

ŁUKASZ KORZENIOWSKI[1] **AND KRZYSZTOF GOCZYŁA**[2]
[1]Nordea Bank Abp SA, 00020 Helsinki, Finland
[2]Faculty of Electronics, Telecommunication and Informatics, Gdańsk University of Technology, 80233 Gdańsk, Poland

Corresponding author: Łukasz Korzeniowski (lukasz.korzeniowski@protonmail.com)

This work was supported by the Gdańsk University of Technology.

ABSTRACT Logging is a common practice in software engineering to provide insights into working systems. The main uses of log files have always been failure identification and root cause analysis. In recent years, novel applications of logging have emerged that benefit from automated analysis of log files, for example, real-time monitoring of system health, understanding users' behavior, and extracting domain knowledge. Although nearly every software system produces log files, the biggest challenge in log analysis is the lack of a common standard for both the content and format of log data. This paper provides a systematic review of recent literature (covering the period between 2000 and June 2021, concentrating primarily on the last five years of this period) related to automated log analysis. Our contribution is three-fold: we present an overview of various research areas in the field; we identify different types of log files that are used in research, and we systematize the content of log files. We believe that this paper serves as a valuable starting point for new researchers in the field, as well as an interesting overview for those looking for other ways of utilizing log information.

INDEX TERMS DevOps, log analysis, logging, knowledge acquisition, system monitoring, reverse engineering, text mining.

I. INTRODUCTION

The need to track a system's behavior during its operation has been a common need since the beginning of software engineering. Traditionally, the main area of focus was failure diagnosis, and the most common form was the recording of actions taken by a system in log files. Studies such as [1] and [2] show that logging is a commonly used practice in the industry. With the rise of cloud computing, new challenges to logging practices have emerged – the distribution of log files among multiple services, a significant increase in log volumes, and a multitude of log formats. At the same time, new opportunities have arisen regarding the potential of the information contained in logs.

One of the rapidly evolving disciplines that explores this potential is log analysis, which strives to discover knowledge from log files (see Fig. 1). The type of knowledge that researchers hope to extract is very broad – from an understanding of system behavior during its operation to drawing conclusions about users' behavior. Log analysis also extends the possibilities in traditional areas of the application of logging data – failure diagnosis and root cause analysis. With a continually growing volume of logs and increasing dispersion of log files across services (especially in cloud environments), conducting a manual analysis becomes very challenging. Commonly used technical solutions for log centralization and aggregation, such as *Splunk* [3] or *LogStash* [4], supported by automated log analysis, can help address these challenges.

The main purpose of this paper is to present an overview of the automated log analysis domain that would serve as a starting point for researchers new to this field. This study is positioned between a systematic mapping study of the domain and a systematic literature review. We identified the most common areas of interest as well as interesting niches based on a systematic review of the recent literature. We split the domain into subfields, focusing on the various types of knowledge that log analysis is capable of extracting. This allows the information potential that lies in the log files to be

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

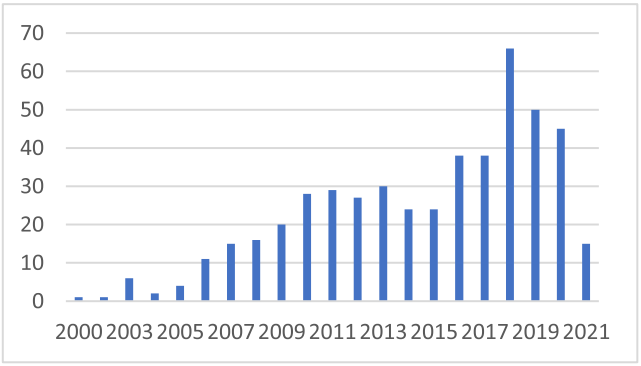

FIGURE 1. Number of papers related to log analysis over the last 20 years, until the middle of 2021.

appropriately presented. Additionally, to support kickstarting in the domain, we provide an overview of different log files and their usage in various applications. Lastly, we collect information about the content that is commonly found or expected to be present in log files, which assists in good orientation in the domain and validation of whether log analysis has the potential to extract the type of knowledge that is of particular interest to researchers. Our review is performed in the context of our research interest in deriving information about the system's structure and behavior during operations using log analysis. Therefore, this area was treated with particular attention in our work. To sum up, our contribution to the field is three-fold:

1) We present an overview of various research areas in the field,
2) We identify different types of log files that are used in research,
3) We systematize the content of log files.

The remainder of this paper is organized as follows. In Section II, we discuss the related work. Section III presents the method we chose to perform the study. Section IV describes the basic assumptions and protocols of the literature review. In Sections V and VI, we present the results of the study, followed by the final conclusions in Section VII. In Appendix 1, we describe the execution of the review according to the defined protocol in detail. The References section contains three types of references: papers mentioned in the article text (references [1] to [7]), papers that were eventually selected for the review after filtering (references [8] to [125]), and papers that were filtered out of the initial set (references [126]–[299]).

II. RELATED WORK

Recently, several reviews related to log analysis have been conducted. [127], [144], and [176] focus on log abstraction – automated methods for generalizing log entries into templates for further analysis. The outcome of log abstraction is log templates, which serve as instructions for log parsers on how to extract meaningful information from a log. Apart from log abstraction, [272] provides a review of research in other log analysis areas, such as failure/anomaly detection and log quality enhancements. The anomaly detection part of [272] (also in the scope of our review) covers the period until 2016, which complements our work. All of the abovementioned papers focus on the technical aspects of logging.

[136] maps the field of failure prediction with correlates with the *Operations/Monitoring* category in our work. The authors identify different types and sources of log files used in this field and identified the limitations and challenges for future research. They point to log formatting and quality issues, log consistency, and the scale, volume, and complexity of logs as the biggest problems. Our work extends this result by providing a content profile for different types of logs.

[129] is another systematic mapping study focusing mostly on the field of log-based software monitoring, which, according to the authors' definition, corresponds to our *Operations* and *Design* areas. In addition to identifying different subfields in this area, the authors also investigate the logging infrastructure and logging practices used by developers. The resulting map of the field is presented from the perspective of the lifecycle of a log. As far as paper selection is concerned, the authors use automated paper filtering in the last stage, which is based on the CORE ranking of conference venues (we perform manual paper filtering based on paper abstracts and/or full text). Because of the different methodologies, focus (log lifecycle vs. knowledge extraction), and date range of the analyzed papers, this research selects different papers for review as compared to our work, and we still find that both works complement each other.

[134] is a recent work reviewing log analysis-related papers with a focus on security (*Operations/Intrusion detection* in our work). The authors take the perspective of research topics (paper keywords). [135] provides a mapping study of methods for linking log entries with the source code that generated them. It summarizes techniques that benefit from log-to-source linkage, as well as classes of problems that are addressed by this approach.

III. STUDY METHOD

We performed our study following a systematic literature review as defined by Kitchenham [5] and [6]. Our process consists of the following phases, which are further elaborated in the following sections:

1) Definition of research questions and a review protocol,
2) Paper search execution and data extraction,
3) Data analysis and providing answers to the research questions.

In the first phase, we defined the research questions that we want to answer. We also described the scope of the study, the inclusion and exclusion criteria for papers, the data source for the research, and the query string used to collect the data. The outcomes of this phase are presented in Section IV. In the second phase, we executed the paper search and

filtered the results according to the defined protocol. We also extracted, analyzed, and synthesized the data obtained from the search query. The details of this process are presented in Appendix 1 and the results are presented in Table 6. Finally, in the third phase, we used the collected data to answer the initially defined research questions and present them in Sections V and VI.

IV. REVIEW PROTOCOL

A. RESEARCH QUESTIONS

To provide an overview of the log analysis domain and some principal information for the new researchers in this field, we want our review to answer the following research questions:

RQ1. What are the different goals of automated log analysis?

RQ2. What common types of log files are used to conduct log analysis?

RQ3. What data attributes can be commonly found in log files?

In the context of our primary research interest (deriving information about the system's structure and behavior from logs), answers to these questions allow us to confirm whether it is a niche worth exploring. They also provide us with a baseline that we can use for performing benchmarks as well as a general overview of data that can be extracted from log files, which we hope will help us in driving our research.

B. INCLUSION AND EXCLUSION CRITERIA

The main driver for our review is research question RQ1, which focuses on the expected outcome of the log analysis processes. Because of this perspective, we include only the papers that clearly describe the effect of log analysis – some valuable information that was collected from log files. At the same time, we exclude papers that focus on the internal mechanics of the process, such as log parsing and improvement of the performance of some algorithms or tools to support the process.

We focus only on automated log analysis, which means that a paper needs to present a consistent, repeatable method for extracting certain information from log files for a particular purpose. We exclude publications that describe manual, ad-hoc analysis that is not repeatable in a different context – approaches whose goal is a one-off retrieval of information to understand a particular phenomenon (e.g., data science papers). In addition, visual analysis utilizing tools to visualize log files and support their analysis, which is based on the user's expertise, is excluded.

We limit our review to the analysis of structured log data. We exclude the analysis of audio/video logs, for example, logs of audio calls in a call center or recordings of video surveillance systems.

Finally, we limit the scope of our review by focusing on primary studies written in English language from the period between 2000 and the first half of 2021. The date range covers the period of greatest interest in the log analysis (see Fig. 1). To keep the number of reviewed papers manageable, we put the biggest focus on the last five years of research. From the 2000-2015 period, we selected only the most cited papers (see Section IV.C for details of this selection).

A summary of the exclusion criteria is presented in Table 1.

TABLE 1. Exclusion criteria.

Criterion number	Criterion name	Excluded papers
EX1	Publication year	Paper published outside of 2016–2021 date range
EX2	Publication language	Papers not written in English
EX3	Primary study	Papers not being primary studies
EX4	Unclear outcome	The outcome of the log analysis presented in the paper is unclear
EX5	Manual log analysis	Papers describing a non-automated approach to log analysis
EX6	Unstructured log data	Papers performing the analysis of unstructured log data
EX7	Technical focus	Papers covering technical aspects of log analysis, e.g., log parsing, support tools for log analysis
EX8	Relevance	Papers not related to automated log analysis

C. DATA SOURCE AND SEARCH QUERY

We use Scopus [7] as the source of papers for our review, which is considered the largest database of abstracts and citations. When constructing a query, we encountered a number of challenges stemming from the fact that *log* is a root word in both Latin and Greek (*logos*). Moreover, it is also a mathematical term, which means that it appears in multiple contexts across multiple fields of science, and consequently returns huge result sets for publication queries. We have also realized that providing a query that precisely applies the earlier defined inclusion/exclusion criteria is nearly impossible – the query would have to be broader and the result set manually filtered. Therefore, we introduced the following criteria when constructing the query:

1. The process of log file analysis is an important aspect for the paper's authors,
2. We focus only on the computer science research area,
3. The result set needs to be manageable within the assumed time and human resources considering the need for manual filtering (no more than 300 papers returned),
4. The fact of information extraction from log files must be explicitly highlighted by the authors of this paper.

The first criterion was met by expecting the article to contain the phrase *log* in its title and the phrase *log analysis* in either the title, abstract, or keywords. The resulting Scopus phrase was TITLE (''log'') AND TITLE-ABS-KEY (''log analysis''). It needs to be pointed out that the keywords included in this phrase cover not only those given by the articles' authors but also keywords automatically indexed by Scopus.

The second criterion was achieved by selecting the *computer science* subject area in the query: LIMIT-TO (SUBJAREA, "COMP").

The third criterion was achieved by analyzing the number of publications over time (see Fig. 1) returned by our query. We decided that limiting the scope of our review to the last five years both matched the defined criteria and covered the period of the biggest interest in log analysis.

In order to meet the last criterion, we referred to the keywords given by the articles' authors, assuming that they have the greatest potential in highlighting the attributes of a paper as seen by its authors. We used the following keywords that indicate the fact that information extraction seems relevant for software systems: *analysis*, *retrieval*, *recovery*, *mining*, *reverse engineering*, and *detection*. The resulting Scopus phrase is as follows:

AUTHKEY ("analysis") OR AUTHKEY ("retrieval") OR AUTHKEY ("recovery") OR AUTHKEY ("mining") OR AUTHKEY ("reverse engineering") OR AUTHKEY ("detection").

The final query that we used was the following:

PUBYEAR > 2015

AND (TITLE("log") AND TITLE-ABS-KEY("log analysis"))

AND (AUTHKEY("analysis") OR AUTHKEY ("retrieval") OR AUTHKEY("recovery") OR AUTHKEY ("mining") OR AUTHKEY("reverse engineering") OR AUTHKEY("detection"))

AND (LIMIT-TO(SUBJAREA, "COMP"))

AND (LIMIT-TO(LANGUAGE, "English"))

To include prominent papers from 2000 to 2015, we applied the same Scopus query for that period and limited the results to papers with at least 20 citations. The threshold for the number of citations may seem to be chosen arbitrarily, but our detailed literature analyzes showed that it is a suitable criterion for selecting notable papers that are at least five years old.

It is important to note that the abovementioned queries precisely apply only the EX1 and EX2 exclusion criteria. The rest of the criteria were applied only roughly and further refined during the manual process described in Appendix 1.

D. THREATS TO VALIDITY

We have identified two major threats to the validity of this study:

1. The scope of papers selected for the review does not cover all of the relevant important papers,
2. The process of manual paper filtering is subject to misinterpretation, which can result in incorrect classification of papers.

Since our research questions are rather broad with the intent of providing an overview rather than a precise answer, it is the size of the paper sample and its representativeness that determines the quality of the answers. Therefore, our mitigation action was to include a broader set of publications while covering the most intensive research period on the subject, even at the cost of manual filtering of papers. The last five-year period was when log analysis was intensively explored; thus, this scope should provide a solid base for providing representative answers to our research questions.

The second threat was mitigated by multiple iterations of the manual classification. For each paper excluded in the manual process, a concrete exclusion criterion was attached together with an argument. Table 7 presents the results of this process.

V. LANDSCAPE OF AUTOMATED LOG ANALYSIS

We provide the answer to RQ1 by presenting the selected papers from the perspective of the goal of log analysis. As all log analysis efforts strive to gain some knowledge, we focus on the different types of knowledge extracted from log files. We identified three types of knowledge that were described in detail in the Section B of Appendix 1 – related to the domain, system design, and system operations. Fig. 2 presents the distribution of selected papers across these categories.

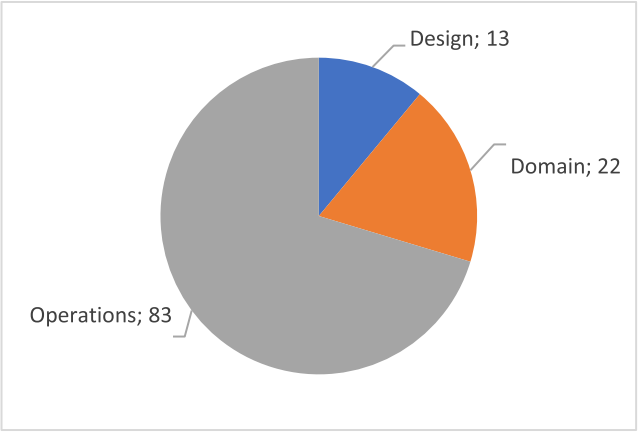

FIGURE 2. Number of papers related to extraction of various types of knowledge through log analysis.

Table 2 summarizes the different application areas that utilize automated log analysis for knowledge extraction. It can be seen that the broadest usage of log analysis takes place in *Software Engineering* and *Cyber-security*. The *Generic* category refers to articles that describe general-purpose log analysis techniques that can be used in multiple areas. Usually, these papers are related to anomaly detection, which is an abstract and generic concept. Two other notable application areas were *Business Process Management* and *E-learning*. It can also be noticed that although automated log analysis is currently being applied mostly in software engineering, the number of different fields that are trying to benefit from such an approach is quite broad, showing several interesting niches for future research.

We further divided the three main types of knowledge into research areas describing the different goals of utilizing the extracted information. Fig. 3 presents this categorization, which we refer to as the *landscape of automated log analysis*. The most prominent research areas and some interesting

TABLE 2. Application areas using automated log analysis.

Application area	Number of papers
Software Engineering	41
Cyber-security	29
Generic	27
Business process management	7
e-learning	3
Manufacturing	2
Online search	1
Online gaming	1
Email	1
Recommendation systems	1
Telecommunications	1
Cyber-Physical behavior	1
User profiling	1
IoT	1
Health	1

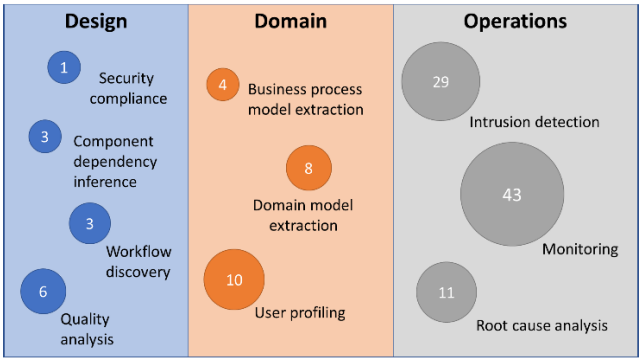

FIGURE 3. Landscape of automated log analysis. Colors denote the type of knowledge extracted from logs; circles are the specific research areas focusing on the given type of knowledge with the number of relevant papers inside.

niches are further described in the subsequent sub-sections. We also introduce the most cited papers (according to Scopus) in each area.

A. OPERATIONS

This type of knowledge relates to information about the running system and constitutes the mainstream of research involving automated log analysis. We further decompose the relevant papers into three research areas: *Monitoring*, *Intrusion detection,* and *Root cause analysis*.

Monitoring refers to activities aimed at watching a running system and detecting situations when it starts to behave unexpectedly. This is an automation of the typical work of system administrators, which focuses on detecting anomalies in observed logs. [76], [32], and [79] present supervised learning, neural network approaches to anomaly detection where logs are encoded into sequences and a sequence machine learning model is applied. [46] additionally addresses the problem of instability of log statements (due to log statement evolution over time or noise introduced by log processing), and [83] focuses on the real-time aspect of anomaly detection. [81] leverages the concept that log statements are in fact not unstructured, as their structure is defined by the source code that outputs them. The authors constructed a control flow using the source code and then matched it with a log file for anomaly detection. Finally, some researchers have focused on anomaly detection specifically in cloud environments. [109] and [61] focus on detecting anomalies within so-called cloud operations, for example, rolling deployments of services into a cloud. [105] touches on the problem of interleaved logs, typical for cloud environments, where multiple task executions create log statements in parallel and log statements need to be automatically mapped to task execution.

Of the earlier (pre-2016) papers, two are notable. [116] is by far the most cited paper in this area. Apart from proposing a method for problem detection using console logs, the authors provide valuable insights related to log processing in general which makes it an especially valuable work regarding any log analysis task. The proposed approach combines source code analysis to determine log patterns and unsupervised machine learning to detect anomalies. [120] focuses on critical infrastructures in which SCADA systems are deployed. The authors propose a method for automated extraction of non-frequent patterns that potentially represent malicious actions.

Intrusion detection is the second most common research area in automated log analysis. It is also related to anomaly detection, but with an explicit focus on the system's security, where each anomaly is treated as a potential threat. Detection of intrusions varies from identification of the fact that the system is under attack to understanding a particular type of attack taking place. [74] used the access log of a web server to distinguish between regular user behavior and malicious scans performed by bots or web crawlers. [39] utilizes attack trees that describe typical sequences of actions for different attack types and matches that information with the content of the log file. [94] dynamically creates anomaly profiles in the form of rules that are further used for attack identification. Some researchers in this field also focus on the detection of particular types of attacks – [103] detects SQL injections, [108] identifies denial of service, and [72] explores the detection of insider threats (those coming from the inside of the protected network).

Some earlier work (pre-2016) needs to be noted. [121] is an interesting approach to intrusion detection in the online gaming domain. The authors detect bot activity by analyzing the individual and collaborative behaviors of players based on game logs. [123] focuses on detecting threats caused by people inside an organization, as opposed to traditionally perceived threats coming from the outside. It uses a probabilistic

approach to detect insiders which strives to maintain a low false alarm rate. [125] explores the area of digital forensics. The authors propose a log model that is later used for the formal analysis and verification of forensic hypotheses based on system logs. They also discuss a real-life example of the usage of their method.

Root cause analysis is a part of bug fixing, the goal of which is to find the core reason for system failure or malfunction. [102] describes an integrated environment for failure detection and root cause analysis based on log files. Correlation analysis is used to identify the root problem. [63] matches system messages stored in a log file with a resource usage log to detect problems related to a lack of resources (e.g., CPU saturation or lack of memory). [20] applied process mining techniques to first reconstruct the process model of the system from its logs and then identify deviations from such a model during process execution. [33] focuses on the analysis of exception logs, mapping them to tasks executed in a cloud environment, and matching them with historical executions of these tasks. [90], [91], and [31] try to identify problems related to specific environments, cloud, and big data platforms (Spark), respectively.

B. DOMAIN

This category is related to the extraction of business knowledge from logs of the software that supports a given domain. The most common research areas in this field are *User profiling*, *Domain model extraction,* and *Business process model extraction*.

User profiling aims to extract knowledge about the structural or behavioral characteristics of users, which support driving further system evolution to better fit users' needs. [89] uses a Hidden Markov Model to extract user intent from actions recorded in logs. [60] explores user intent in a cyber-physical context. It matches user actions in cyberspace (by analysis of web query logs) with the user's physical location (WiFi access point logs) to understand and predict their behavior in the physical world. [101] captures an exploration of the user behavior into a higher-level concept of usage tactics, which, according to the authors, allows for better interpretability and comparability between systems. [95] extracts the structural profile of users to provide product recommendations. It focuses on new (previously unknown) users without any shopping history, for whom it utilizes an access log to derive the user's interests and suggest suitable products.

Of the earlier studies in this area, two are notable. [119] seeks to discover the actual user intent (a subtask that user wants to fulfil) by analyzing the query entered in a search engine together with the corresponding links that were clicked afterwards and additional refining keywords entered in subsequent searches. The authors of [116] use client and server logs capturing user's interactions with a website to build a user profile. The intention is to use such user profiles to personalize the user interface of web applications for specific users.

Domain model extraction refers to understanding some real-life (domain) phenomena using information from log files. [107] uses an anonymized web search query log to identify adverse drug reactions. [70] and [68] explore the educational domain. [70] aims to understand the correlation between students' performance and students' behavior, and their tutor's teaching style. [68] is a boundary paper between domain model extraction and user profiling, which models students' behavior using the Hidden Behavior Traits Model. The authors of [64] learn expert knowledge on applying security rules from computers secured by professionals and apply this knowledge to previously unseen systems of non-experts. This paper treats the security log as a carrier of hidden domain knowledge. [35] uses process mining techniques to discover the ontology of the computer science domain.

Apart from the abovementioned work, there is also some prominent research available from the earlier period that explores the concept that observation of how people search through the Internet allows us to discover their goals or to better understand the topic they are searching for. The earliest paper in this area is [113] which utilizes search engine logs for the categorization of search query terms into a predefined taxonomy. [114], the most cited paper in this field, uses both search engine logs and actual user clicks that follow the search to explain the semantic relationships between search queries. The results are presented in the form of query graphs.

Business process model extraction aims to understand business processes from the log of system actions. [66] uses a frequent itemset mining approach to extract knowledge about the business process from an event log. [67] considers how the level of abstraction of a business process extracted from logs influences conformance with the actual process, which is crucial to balance process abstraction and accuracy. [9] focuses on the detection of anomalies in the event log using the model-agnostic approach, where no reference process model is available. It aims to provide a method for cleaning the event log, which would result in increased accuracy of the derived process model.

[117] is a notable earlier work that uses workflow logs to recreate the actual business process realized by an application and to compare it with the anticipated process. According to the authors, such an approach allows for optimizing business processes especially in terms of applying error handling more precisely which should result in lowering the process-modeling cost.

C. DESIGN

The design category relates to extracting knowledge about the internal workings of a system (e.g., system structure), software building process, or attributes related to its design (e.g., quality or security). We split this category into four research areas: *Quality analysis*, *Workflow discovery*, *Component dependency inference,* and *Security analysis*.

The *Quality analysis* research area groups papers that refer to the assessment of system quality. [29] uses information from the log files of a running system to reconstruct

production-like workloads for further use during system testing. Additionally, the authors analyzed the representativeness of the recovered workloads based on the varying levels of granularity of user actions considered for the recovery process. [93] applies a similar approach of exploring typical user interactions with a system to construct a test that assesses the reliability of the system. The authors used the mean time between failures as a measure of the system's reliability and validated their approach against a real-life system. [54] and [16] focus on the quality of SQL queries in the analyzed software. They analyzed the log of SQL queries executed by a system and detected anti-patterns.

[124] is a notable earlier work (pre-2016) that attempts to explain the usability characteristics of an application by analyzing search queries entered by users in a web browser regarding that application. Such an approach allows to gather user feedback regarding both the existing and the desired functionality of an application.

The *Workflow discovery* research area is related to the discovery of internal software processes. [65] describes a process mining approach that can discover recursive processes from event logs. [92] reconstructs workflows (series of interactions between services) in a cloud environment with a focus on failed workflows. Additionally, [122] is a widely cited work from 2014 that recovers a Communicating Finite State Machine model of concurrent system behavior. The approach presented by the authors is capable of utilizing any log file but requires users to provide a set of regular expressions to extract the expected pieces of information from log lines.

Component dependency inference captures papers that aim to recover the internal dependencies between software components (services). [98] uses service logs to identify the composition and substitution relationships between services constituting a software system. [67] uses predictive and statistical analyses of web service invocations from service logs to identify the relationships between services. The authors also propose a classification of the types of dependencies between services. Out of the pre-2016 papers, [118] is the most cited in this area. It uses Bayesian Decision Theory to infer dependencies between components in a distributed system and validates this approach against the Hadoop MapReduce framework.

[42] focuses on *Security compliance* and explores the compliance of the assumed security rules with their actual effect. The authors propose a method for automated analysis of the access log to detect conflicting security rules.

VI. TYPES OF LOGS USED IN RESEARCH AND THEIR CONTENT

Research questions RQ2 and RQ3 are related to the classification and content of log files commonly used in research. In the Section C of Appendix 1, we define classes of log files, and Table 3 lists their occurrence in various areas of research. It can be seen that the three most commonly used types of log files are: *Generic*, *Proprietary,* and *Network*. The strongest correlation can be observed between the *Generic* log class and *Operations* research, and more specifically, the *Anomaly detection* category, which abstracts from the concrete log format.

TABLE 3. Types of logs used to extract different types of knowledge.

Log type	Design	Domain	Operations	Sum
Access log		2	7	**9**
CD log			2	**2**
Event log		4	2	**6**
Generic	6	4	32	**42**
Network log			11	**11**
Platform log		1	8	**9**
Proprietary	4	7	21	**32**
Query log	2			**2**
Search engine log	1	4		**5**

It can be also observed that if we take away the *Generic* log type, *Proprietary* logs are by far the most used for analysis in research. This suggests the lack of standardization of log files and shows the need to explore the common properties of these logs. Table 4 presents a statistical summary of the contents of the various types of logs. Green-colored columns present the number of occurrences of each attribute class in the papers reporting the usage of a given log type. The color intensity visualizes how common each attribute class is within a given log type.

The last column summarizes the ubiquity factor of the log attribute classes, which is defined in detail in the Section D of Appendix 1. The ubiquity value is [0,1] normalized and represents how common the given attribute class is across all log types reported in the selected papers.

Table 4 allows the creation of a statistical profile for each log type. The statistics are gathered based on the log attributes reported in the selected papers, which, depending on a paper, are a mixture of full log contents and only those attributes that the authors found useful for their log analysis. This means that the values presented in Table 4 embed both the availability and usefulness factors for each log attribute class. *Access*, *Event,* and *Query* logs are either well-defined log types (access log) or strongly embedded in a particular field or method (event log – process mining, query log – SQL analysis). Therefore, their profile represents the actual log format specification or the requirements of the technique used. *Generic*, *Network*, *Platform,* and *Proprietary* log types are non-standardized, which makes their profiles more interesting. The *Platform* log exhibits *Resource use* information as the most commonly used attribute class, while *Event* is the most frequent in the others. The *Network log* focuses on the *Source*, *Destination,* and *Data size* classes, which are related to the network traffic being tracked by them. All non-standard log files contain *Timing* as important information.

If we take the attribute class perspective, the ubiquity factor column in Table 4 presents an average statistical profile of a

TABLE 4. Presence of log attribute classes in different types of logs.

Log attribute class	Access log	Event log	Generic	Network log	Platform log	Proprietary	Query log	Search engine log	Ubiquity
Action	23	0	1	4	1	6	2	3	0.39
Authentication information	0	0	0	0	0	4	0	0	0.01
Communication channel	0	1	0	3	0	2	0	0	0.02
Component	0	1	4	1	5	4	0	0	0.10
Data size	11	0	0	8	0	3	0	0	0.09
Destination	11	0	2	12	4	9	0	1	0.32
Event	11	10	9	37	7	29	0	0	0.86
Log file information	0	0	3	0	0	1	0	0	0.01
Object	0	6	3	11	5	5	0	0	0.21
Resource use information	0	0	0	0	23	4	0	0	0.07
Severity	0	0	2	3	1	2	0	0	0.04
Source	0	1	0	11	2	4	0	0	0.10
Timing information	0	1	5	8	12	18	0	1	0.37
User information	22	0	0	3	3	6	0	0	0.19

log across all log types. In general, it can be seen as the chance of finding a given attribute class in a log. The average log profile consists of (in order of decreasing ubiquity): *Event*, *Timing information*, *Action*, *Destination*, *Object,* and *User information*.

VII. CONCLUSION

We performed a systematic literature review and a mapping study of the automated log analysis research area since 2000 until halfway through 2021, with the main focus on the last five years. We mapped the area into sub-fields from the perspective of the type of knowledge that can be extracted from different log files and the goal of such an analysis. We presented the results in the form of the landscape of automated log analysis, characterizing each subfield and introducing the most prominent recent research. Additionally, we performed an in-depth analysis of log files and summarized the different types of logs commonly used in research, together with their content. We have provided a statistical profile of each log type, which allows researchers to better understand what type of information is expected to be available in various logs. Additionally, we made all source information that was the basis for our analysis available in the form of appendices.

We hope that our work will be valuable to researchers and practitioners who aim to explore the challenging idea of extracting knowledge on complex, sometimes hard to manage, computer systems from the system logs.

In our future work, we will focus our research on the *Component dependency inference*, which seems to be fairly unexplored area. Our main interest lies in the assessment of the capabilities of log analysis to extract knowledge about software components and processes that govern them.

APPENDIX 1 – REVIEW PROCESS EXECUTION

We executed the review according to the defined protocol in three phases. First, we executed the defined query and applied the exclusion criteria. The outcome of this phase was a list of relevant papers that were used in the subsequent steps. In the second phase, we extracted features to support answering the research questions, while in the third phase, we synthesized these features. The subsequent sections describe each phase in more detail. For clarity, the feature extraction and synthesis phases are discussed separately for each research question.

A. PAPER FILTERING

Execution of the final query in the Scopus database on 30.06.2021 returned 292 papers. The exclusion criteria EX1 and EX2 were already embedded in the query. For each paper from the result dataset, we applied the following multistep process:

1. Filter out not relevant papers based on abstracts,
2. Apply exclusion criteria using the paper's abstract,
3. If the paper cannot be clearly excluded based on its abstract, apply the exclusion criteria using the full text,
4. Remove duplicates.

The first step is necessary because of the assumed strategy for paper selection; as the query is not precise enough, it can retrieve papers that are not relevant to log analysis. We were able to identify all such papers using only their abstracts.

We used the second step to reduce workload during the application of the exclusion criteria. We applied only exclusion criteria EX5, EX6, EX7, EX8, and EX3 at this stage. To avoid falsely excluded papers, we used a defensive approach and omitted the application of EX4. In the cases where the abstract of some of a paper did not provide enough

evidence to exclude it based on the abstract, we qualified such a paper for the next step.

After initial filtering based on abstracts, for each paper that was not excluded, we applied the exclusion criteria based on the paper's full text. We focused on exclusion criteria EX4, EX5, EX7, and EX8 and searched for evidence justifying their application. After completing this process, as part of exclusion criterion EX3, we removed duplicate papers. The set of selected papers after the filtering process consisted of 118 publications.

Table 5 summarizes the papers excluded. The main reason for excluding articles was their technical focus – not covering direct methods for extracting knowledge from logs, but focusing rather on tools and algorithms supporting this process (e.g., log parsing, template generation, or log visualization). Another commonly excluded category of papers was those describing manual log analysis. Although our work focuses on automatic approaches, the excluded papers often present interesting ideas on utilizing logs for gathering domain knowledge. These approaches have the potential for automation, which could make them fall under the scope of automated log analysis in the future. The third most common exclusion criterion was a lack of clarity. We used this category if the paper's abstract was not clear enough on the outcome of the log analysis, and the full text was not available. We also used it to mark preliminary work or experience reports that did not describe a concrete result. A summary of the excluded papers, together with the exclusion criteria applied and the justification, is presented in Table 7.

TABLE 5. Summary of excluded papers.

Criterion number	Criterion name	Number of excluded papers
EX7	Technical focus	76
EX5	Manual log analysis	51
EX4	Unclear outcome	29
EX3	Primary study	9
EX6	Unstructured log data	5
EX8	Relevance	4

B. RQ1 – FEATURE EXTRACTION AND SYNTHESIS

We collected the following information from each paper:

- Goal of the log analysis,
- Business area/application domain.

Such a choice of attributes allows the presentation of various research areas within log analysis from both technical and business perspectives. For each paper, we extracted the data by looking into the paper's title, authors' keywords, and finding additional evidence supporting this selection in the full text of the paper. We further classified the papers according to the type of extracted knowledge, which was further subdivided into research areas. We define the following types of knowledge:

- Domain – knowledge about a business domain, for example, improved understanding of business processes, or understanding of user behavior,
- Design – knowledge related to a software system and the process of its design, for example, understanding the relationships between components, or detecting system quality issues,
- Operations – knowledge related to the running system during operation, for example, detecting anomalies in the system's behavior, or predicting the system's failure.

Detailed data on the classification of each paper are presented in Table 6.

C. RQ2 – FEATURE EXTRACTION AND SYNTHESIS

The type of log file was extracted from the full text of the publications. We searched for named types of logs or information that a proprietary log file was used for the research. In some cases, the study used a generic model of a log. We synthesized the various log types used in the papers into the following classes:

- Access log – server log recording HTTP requests,
- CD log – log of continuous engineering tools (continuous integration/continuous deployment),
- Event log – log of business events, used by process mining techniques,
- Generic – log format is automatically detected using the technique described in the paper, or the paper assumes some log model,
- Network log – log of a network device or service (e.g. SSH, proxy, firewall),
- Platform log – log of a specific software platform (e.g. Spark, Hadoop, Android),
- Proprietary – log of a particular software system, in a custom format that cannot be classified into other classes,
- Query log – log of SQL queries executed by a system,
- Search engine log – log of a search engine consisting of search queries entered by a user.

A detailed classification of the log types for each included paper is presented in Table 6.

D. RQ3 – FEATURE EXTRACTION AND SYNTHESIS

To extract the various data attributes that can be found in log files, we again referred to the full text of the article, searching either for a named type of log file or an enumerated list of attributes used in that particular research. Named types of logs often represent a well-established log standard in a given area that is publicly described. In such cases, we derived the data attributes from the formal definition of the log file. We classified the identified attributes into the following classes that represent the different types of information represented by each attribute:

- Action – information related to a recorded user/client action,
- Authentication information – information related to a user's/client's credentials,
- Communication channel – information related to a channel on which a communication that was recorded as log entry was established,

TABLE 6. Knowledge extracted from papers.

References	Goal of log analysis	Business area/application domain	Type of log used	Type of extracted knowledge	Research area	Unified log type
[9]	Anomaly detection	Business process management	Log model	Domain	Business process model extraction	Generic
[24]	Anomaly detection	Cyber-security	Proprietary	Operations	Intrusion detection	Proprietary
[73]	Anomaly detection	Cyber-security	Access log	Operations	Intrusion detection	Access log
[10], [11], [94]	Anomaly detection	Cyber-security	Generic	Operations	Intrusion detection	Generic
[84]	Anomaly detection	Generic	Access log	Operations	Monitoring	Access log
[15], [18], [27], [28], [36], [44], [45], [46], [48], [49], [50], [51], [79], [81], [83], [85]	Anomaly detection	Generic	Generic	Operations	Monitoring	Generic
[76]	Anomaly detection	Generic	LANL dataset	Operations	Monitoring	Proprietary
[111]	Anomaly detection	Generic	network logs	Operations	Monitoring	Network log
[82]	Anomaly detection	Generic	network system logs	Operations	Monitoring	Network log
[88]	Anomaly detection	Generic	Proprietary	Operations	Monitoring	Proprietary
[56]	Anomaly detection	Software engineering/HPC	Generic	Operations	Monitoring	Generic
[62]	Anomaly detection	Software engineering	Generic	Operations	Monitoring	Generic
[91]	Anomaly detection	Software engineering/Big data	Spark log	Domain	Root cause analysis	Platform log
[61]	Anomaly detection	Software Engineering/Cloud	Cloud operations log	Operations	Monitoring	Proprietary
[90]	Anomaly detection	Software engineering/Cloud	Kubernetes logs	Operations	Root cause analysis	Platform log
[109]	Anomaly detection	Software engineering/Cloud	Proprietary	Operations	Monitoring	Proprietary
[52]	Anomaly detection	Software engineering/Networking	DNS server logs	Operations	Monitoring	Network log
[53]	Anomaly detection	Software engineering/Networking	Proprietary	Operations	Monitoring	Proprietary
[57]	Anomaly detection	Telecommunications	Proprietary	Operations	Monitoring	Proprietary
[120]	Anomaly detection	Manufacturing	Proprietary	Operations	Monitoring	Proprietary
[121]	Anomaly detection	Online gaming	Game log	Operations	Monitoring	Proprietary
[54]	Antipattern detection	Software Engineering/Quality	Query log	Design	Quality analysis	Query log
[16]	Antipattern detection	Software Engineering/Quality	Query log	Design	Quality analysis	Query log
[89]	Behavior analysis	Business process management	Event log	Domain	User profiling	Event log
[60]	Behavior analysis	Cyber-Physical behavior	Access point association log, web query log	Domain	User profiling	Proprietary
[22]	Behavior analysis	Cyber-security	Physical access log	Operations	Intrusion detection	Proprietary
[69]	Behavior analysis	e-learning	Moodle logs	Domain	User profiling	Proprietary
[68]	Behavior analysis	e-learning	Proprietary	Domain	Domain model extraction	Proprietary
[12]	Behavior analysis	Generic	Access log	Operations	Monitoring	Access log
[107]	Behavior analysis	Health	Search engine log	Domain	Domain model extraction	Search engine log
[21]	Behavior analysis	Manufacturing	Abstract log model	Domain	User profiling	Generic
[101]	Behavior analysis	Online search	User action log	Domain	User profiling	Proprietary

TABLE 6. *(Continued.)* **Knowledge extracted from papers.**

References	Goal of log analysis	Business area/application domain	Type of log used	Type of extracted knowledge	Research area	Unified log type
[95]	Behavior analysis	Recommendation systems	Access log	Domain	User profiling	Access log
[58]	Behavior analysis	Software engineering/Mobile	Application logs, event logs	Design	Quality analysis	Proprietary
[59]	Behavior analysis	Software Engineering/Web	Access log	Domain	User profiling	Access log
[77]	Business process management	Business process management	Event log	Domain	Domain model extraction	Event log
[17]	Debugging hints	Software Engineering/Bug fixing	Tizen platform logs	Operations	Root cause analysis	Proprietary
[68], [98]	Dependency inference	Software engineering/Reverse-engineering	Proprietary	Design	Component dependency inference	Proprietary
[118]	Dependency inference	Software engineering/Reverse-engineering	Generic	Design	Component dependency inference	Generic
[122]	Dependency inference	Software engineering/Reverse-engineering	Generic	Design	Workflow discovery	Generic
[92]	Discover interactions between services	Software engineering/Cloud	Log model: Behavior Log, Event log	Design	Workflow discovery	Generic
[64]	Domain model extraction	Cyber-security	Microsoft event log	Domain	Domain model extraction	Platform log
[113], [114]	Domain model extraction	Generic	Search engine log	Domain	Domain model extraction	Search engine log
[108]	DOS attack detection	Cyber-security	Network logs	Operations	Intrusion detection	Network log
[13]	Error resolution	Software Engineering/Bug fixing	Configurable	Operations	Root cause analysis	Proprietary
[30]	Failure detection	Software engineering/HPC	Resource use log, message log	Operations	Monitoring	Proprietary
[47]	Failure detection	Software engineering	Network logs	Operations	Monitoring	Network log
[20]	Failure detection	Software Engineering/Bug fixing	Generic	Operations	Root cause analysis	Generic
[102]	Failure detection	Software engineering/Cloud	Configurable	Operations	Root cause analysis	Proprietary
[33]	Failure detection	Software engineering/Cloud	system log	Operations	Root cause analysis	Platform log
[32]	Failure prediction	IoT	Generic	Operations	Monitoring	Generic
[37]	Intrusion detection	Cyber-security	Network logs	Operations	Intrusion detection	Network log
[8]	Intrusion detection	Cyber-security	SSH logs	Operations	Intrusion detection	Network log
[25]	Intrusion detection	Cyber-security	TOR log	Operations	Intrusion detection	Network log
[41], [74], [110]	Intrusion detection	Cyber-security	Access log	Operations	Intrusion detection	Access log
[72]	Intrusion detection	Cyber-security	Carnegie Mellon University's Insider Threat Program dataset	Operations	Intrusion detection	Proprietary
[86]	Intrusion detection	Cyber-security	Configurable	Operations	Intrusion detection	Generic
[40], [106]	Intrusion detection	Cyber-security	Proprietary	Operations	Intrusion detection	Proprietary
[71]	Intrusion detection	Cyber-security	proxy logs, Honeyclient logs	Operations	Intrusion detection	Network log
[123]	Intrusion detection	Cyber-security	System log	Operations	Intrusion detection	Platform log
[125]	Intrusion detection	Cyber-security	Log model	Operations	Intrusion detection	Generic
[39]	Intrusion detection	Cyber-security/Mobile	Android logs	Operations	Intrusion detection	Proprietary
[103]	Intrusion detection	Cyber-security/Web	Proprietary	Operations	Intrusion detection	Proprietary
[97]	Intrusion detection	Software	Access log	Operations	Intrusion	Access log

TABLE 6. *(Continued.)* **Knowledge extracted from papers.**

References	Goal of log analysis	Business area/application domain	Type of log used	Type of extracted knowledge	Research area	Unified log type
		engineering/Web			detection	
[29]	Load test design	Software engineering/Quality	User action log	Design	Quality analysis	Proprietary
[100]	Malware detection	Cyber-security	Event log	Operations	Intrusion detection	Event log
[23]	Malware detection	Cyber-security	proxy logs	Operations	Intrusion detection	Network log
[14]	Monitoring	Software Engineering	Generic	Operations	Monitoring	Generic
[19]	Performance prediction	Software engineering/HPC	HPC I/O logs	Operations	Monitoring	Platform log
[80]	Problem diagnosis	Generic	Generic	Operations	Monitoring	Generic
[26]	Problem diagnosis	Software engineering/Continuous Engineering	CD logs	Operations	Monitoring	CD log
[115]	Problem diagnosis	Software engineering	Generic	Operations	Monitoring	Generic
[66]	Process discovery	Business process management	Event log	Domain	Business process model extraction	Event log
[65]	Process discovery	Business process management	Log model	Design	Business process model extraction	Generic
[117]	Process discovery	Business process management	Generic	Design	Business process model extraction	Generic
[35]	Process mining	Generic	Event log	Domain	Workflow discovery	Event log
[99]	Process optimization	Software engineering/Big data	Hadoop storage-tier logs	Operations	Monitoring	Platform log
[96]	Process optimization	Software engineering/Cloud	Proprietary	Operations	Monitoring	Proprietary
[104]	Process performance assessment	Business process management	Event log	Operations	Monitoring	Event log
[93]	Quality assurance	Software Engineering	Generic	Design	Quality analysis	Generic
[124]	Quality assurance	Software Engineering	Online search query log	Design	Quality analysis	Search engine log
[31]	Root cause analysis	Software Engineering/Big data	Spark log	Operations	Root cause analysis	Platform log
[63]	Root cause analysis	Software engineering	Resource use log, message log	Operations	Root cause analysis	Platform log
[34]	Root cause analysis	Software engineering/Cloud	Generic	Operations	Root cause analysis	Generic
[55]	Root cause analysis	Software Engineering/Continuous Engineering	CD logs	Operations	Root cause analysis	CD log
[42]	Security Analysis	Cyber-security	Access log model	Design	Security analysis	Generic
[75], [87]	Security Analysis	Cyber-security	Generic	Operations	Intrusion detection	Generic
[112]	Security Analysis	Cyber-security	network logs	Operations	Intrusion detection	Network log
[38]	Spam detection	Email	Email reception logs	Operations	Intrusion detection	Proprietary
[70]	Student performance prediction	e-learning	Proprietary	Domain	Domain model extraction	Proprietary
[43]	User profiling	User profiling	Proprietary	Domain	User profiling	Proprietary
[116]	User profiling	Software engineering/Web	Proprietary	Domain	User profiling	Proprietary
[119]	User profiling	Software engineering/Web	Access log, Search engine log, Click log	Domain	User profiling	Search engine log
[105]	Workflow monitoring	Software engineering/Cloud	Log model	Operations	Monitoring	Generic

- Component – information about a software component/module that the log entry is related to,
- Data size – information related to the size of data processed/transferred as a result of executing an action,
- Destination – target (host/system/component) of a recorded communication event,
- Event – details of a recorded event (usually a message text),

TABLE 7. Excluded papers.

Reference	Applied exclusion criterion	Justification
[126]	EX6	Audio log
[127]	EX3	Review
[128]	EX4	No full text available
[129]	EX3	Systematic mapping study
[130]	EX7	Log grouping
[131]	EX4	No full text available
[132]	EX7	Evaluation of tool
[133]	EX7	Support tool for log analysis
[134]	EX3	SLR
[135]	EX3	Systematic mapping study
[136]	EX3	Systematic mapping study
[137]	EX7	Support tool for log analysis
[138]	EX8	Related to event streams rather than log files
[139]	EX7	Log feature extraction
[140]	EX7	Support tool for log analysis
[141]	EX5	Visual analysis of logs
[142]	EX7	Dimensionality reduction
[143]	EX7	Dataset of CI/CD logs
[144]	EX3	SLR
[145]	EX7	Log parsing
[146]	EX5	Gain understanding on the common bug-fixing process
[147]	EX4	No full text available
[148]	EX5	Manual log analysis based on results presented by a support tool
[149]	EX5	Visual analysis of logs, no full text
[150]	EX5	Analysis of logs from educational tasks
[151]	EX5	Manual analysis of network attacks to improve infrastructure security
[152]	EX7	Support tool for log analysis
[153]	EX7	Representation of time in graph database
[154]	EX4	No full text available
[155]	EX4	No full text available
[156]	EX7	Comparison of tools for log processing
[157]	EX5	Manual analysis of SPARQL query structure
[158]	EX5	Describes the capabilities of log analysis with a real-life example
[159]	EX5	Manual analysis of data collected from logs
[160]	EX4	No full text available
[161]	EX4	No full text available
[162]	EX6	Audio log
[163]	EX7	Log management
[164]	EX4	No full text available
[165]	EX5	Data science
[166]	EX4	No full text available
[167]	EX5	Visual log analysis
[168]	EX7	Support tool for log analysis
[169]	EX7	Preserving anonymity during log analysis
[170]	EX7	Performance of log analysis over large log datasets
[171]	EX7	Role of silver-data in machine learning
[172]	EX4	No full text available
[173]	EX5	Analysis of properties of Fintech log files
[174]	EX7	Log file parsing
[175]	EX5	Analysis of e-journal usage
[176]	EX3	Review of log parsers
[177]	EX8	Well logging
[178]	EX8	Designing of log format for healthcare
[179]	EX7	Log pattern mining
[180]	EX6	Video logs
[181]	EX7	Log parsing
[182]	EX7	Graph-database representation of log files
[183]	EX7	Support tool for log analysis
[184]	EX7	Support tool for log analysis

TABLE 7. *(Continued.)* Excluded papers.

Reference	Applied exclusion criterion	Justification
[185]	EX7	Support tool for log analysis
[186]	EX7	Extension of a support tool for log analysis
[187]	EX4	No full text available
[188]	EX5	Data science
[189]	EX4	No full text available
[190]	EX7	Verification of suitability of log clustering in continuous engineering
[191]	EX5	Experiment summary
[192]	EX5	Case study
[193]	EX5	Data science
[194]	EX7	Support tool for log analysis
[195]	EX4	No full text available
[196]	EX7	Support tool for log analysis
[197]	EX4	Tool proposal
[198]	EX7	Support tool for log analysis
[199]	EX7	Support tool for log analysis
[200]	EX4	No full text available
[201]	EX7	Support tool for log analysis
[202]	EX7	Support tool for log analysis
[203]	EX7	Comparison of activation functions
[204]	EX7	Support tool for log analysis
[205]	EX5	Manual log analysis of NDN network logs
[206]	EX7	Support tool for log analysis
[207]	EX7	Support tool for log analysis
[208]	EX7	Support tool for log analysis
[209]	EX7	Support tool for log analysis
[210]	EX7	Design of a support tool for log analysis
[211]	EX7	Support tool for log analysis
[212]	EX5	Manual analysis using a tool
[213]	EX4	Experience report
[214]	EX7	Evaluation of a method
[215]	EX7	Comparison of tools for log processing
[216]	EX6	Audio log
[217]	EX4	No full text available
[218]	EX4	Preliminary work
[219]	EX5	Data science
[220]	EX6	Image/sensor logs
[221]	EX7	Log burst detection
[222]	EX7	Support tool for log analysis
[223]	EX7	Support tool for log analysis
[224]	EX7	Log parser
[225]	EX7	Support tool for log analysis
[226]	EX5	Data science
[227]	EX7	Support tool for log analysis
[228]	EX7	Support tool for log analysis
[229]	EX7	Template detection
[230]	EX5	Data science
[231]	EX5	Data science
[232]	EX7	Log compression
[233]	EX7	Support tool for log analysis
[234]	EX7	Evaluation of SIEM tool extension
[235]	EX5	Data science
[236]	EX7	Evaluation of support tools
[237]	EX7	Support tool for log analysis
[238]	EX4	No full text available
[239]	EX4	No full text available
[240]	EX5	Data science
[241]	EX5	Data science
[242]	EX5	Data science
[243]	EX5	Data science
[244]	EX4	No full text available
[245]	EX4	No full text available
[246]	EX7	Support tool for log analysis
[247]	EX7	Event log pre-processing
[248]	EX7	Support tool for log analysis
[249]	EX5	Data science
[250]	EX5	Data science
[251]	EX7	Support tool for log analysis
[252]	EX3	Literature survey

TABLE 7. *(Continued.)* **Excluded papers.**

Reference	Applied exclusion criterion	Justification
[253]	EX5	Manual analysis of LogCat logs
[254]	EX7	Audit data collection for log analysis
[255]	EX7	Support tool for log analysis
[256]	EX7	Tool for synthetic log generation
[257]	EX7	Support tool for log analysis
[258]	EX5	Manual analysis with ELK
[259]	EX7	Scalability of log analysis
[260]	EX7	Method for log coding
[261]	EX5	Data science
[262]	EX7	Evaluation of tool
[263]	EX4	No full text available
[264]	EX4	No full text available
[265]	EX7	Support tool for log analysis
[266]	EX5	Data science
[267]	EX5	Data science
[268]	EX4	No full text available
[269]	EX4	No full text available
[270]	EX4	No full text available
[271]	EX4	No full text available
[272]	EX3	SLR
[273]	EX4	Preliminary work
[274]	EX5	Manual analysis using a tool
[275]	EX5	Data science
[276]	EX5	Data science
[277]	EX5	Data science, demonstration of technology capabilities
[278]	EX5	Data science
[279]	EX7	Support tool for log analysis
[280]	EX7	Evaluation of tool
[281]	EX5	Data science
[282]	EX5	Data science
[283]	EX5	Data science
[284]	EX5	Data science
[285]	EX5	Data science
[286]	EX5	Data science
[287]	EX5	Data science
[288]	EX5	Data science
[289]	EX7	Method for message type extraction
[290]	EX5	Data science
[291]	EX7	Evaluation of time coalescence techniques
[292]	EX5	Data science
[293]	EX7	Evaluation of tool
[294]	EX5	Data science
[295]	EX7	Method for pattern mining
[296]	EX7	Evaluation of usefulness of dimensionality reduction techniques in anomaly detection
[297]	EX8	Not related to log analysis
[298]	EX7	Method for user session clustering
[299]	EX7	Method for user session clustering

- Log file information – information about the file in which the log entry was created,
- Object – information about the destination system's business object that is the subject of a recorded event,
- Resource use information – information related to the utilization of a system's resources,
- Severity – information about the importance of a recorded event,
- Source – source (host/system/component) of a recorded communication event,

TABLE 8. **Unification of log attribute names.**

Unified log attribute name	Original log attribute name
Action	SQL query, ErrorCode, failure code, HTTP status code, STATUS, status code, ErrorMessage, destination URLs, Data require location, HTTP query, location, resource path, URL, web service name, type of action, action details
Authentication information	Authentication orientation, Authentication type, Logon type, Success/failure
Communication channel	channel, Protocol, request protocol, request_protocol
Component	component, job, process, ProgName, service name, class name, event recorder, function name, log module, related class, server module, server_module, thread ID, location (Object path, Source, SCADA node)
Data size	byte, Bytes delivered, Bytes queued, data size, Data volume sent/received, FileSize, LEN, Length, Messages delivered, Messages queued, number of request/response bytes, number of source/destination packets
Destination	Destination, destination IP Address, destination IP/Port, Destination pc, Destination user, Destination IP, Destination port, DST, dstIP, dstIP, To, ClientIP, details of the invoked server, host, Host ID, hostName, ip, IP/Port, ipAddress with sub-properties ip4Address and ip6Address, machine name, node, Remote host, server target, server_target, server_URL
Event	authentication results of SPF, application & service-specific objects, attributes, Average number of accessing the module per week [0-10 times], Average time delay in accessing the lectures starting from the upload time [0-100 days], Average time delay in accessing the lectures starting from the upload time[0-100 days], Average time of uploading the assignment answers subtracted from deadline time [0-100 days], city, Classification, Control address, customer indicators, Daemon, decision, Delay in enrolment to the module [0-100 days], device type, DF, Direction, DKIM, DKIM signature domain, DPT, DSN, Envelope-From domain, Header-From domain, Institution code, Number of accessing the module and resources in the semester [0-100 times], other data, PHYSIN, PHYSOUT, PREC, Project File Structure, province, RES, seq, SPT, SRC, Stat, tags, TOS, TTL, Tty, URGP, WINDOW, action, activity, event type, HTTP method, message type, operation, RequestType, type of passage (in, out, empty), type of request, user action, content, description, Error Message, event description, event details, External Error, Info, log message, logMessage, message, message body, message text, Msg, name of order, system and business faults, Systemmessage, Traceback And Nested Errors, virus name, PID, Source Code Line and File, name of attack (URL), load, unload, click, mouseover, mouseout
Log file information	line number in log file, log type, logFilePath, LogType
Object	Access point Id, case id, CVEID, device ID, device key, EUID, event id, Executor ID, Id, Message ID, object id, object identifier, Queue ID, request id, request_session, router card number, Rule number, Stage ID, Task ID, traceNo, transaction ID, UID, name of attack, program object, Relay, Rule
Resource use information	[Read/Write]BytesTotal, [Read/Write]Less1M, [Read/Write]Throughput, After Heap GC space, After Young GC space, Before GC Young space, Before Heap GC space, Consec[Read/Write]Pct, Full GC time, GC category, GC time, GroupID, Heap

TABLE 8. *(Continued.)* **Unification of log attribute names.**

	space, number of calls, NumFile, NumNodes, NumProcs, resource use counters, response time, Seq[Read/Write]Pct, server url, StripeCount, StripeSize, Total[Read/Write]Req, TotalMetaReq, user time (CPU)
Severity	level, log level, logging level, Priority, Severity
Source	departure, departure IP, Event source, From, Sender, Source, source file, source IP, Source pc, Source user, SourceIP, Source port, srcIP
Timing information	Application execution time, duration of the related operation, Job execution time, Stage execution time, Task execution time, Association duration, request session, session lifetime, Association start timestamp, Date, event registration timestamp, infection time, invocation date and time, real time, request time, request_time, sys time, Time, TimeCreation, times, Timestamp
User information	HTTP user agent, Mailer, remote user, user, user account, user agent, user role, user_role, UserName

- Timing information – information related to the time that a recorded event took place and its duration,
- User information – information related to the user that a recorded event is related to.

Details of the classification of each attribute identified in the selected papers are presented in Table 8.

For each attribute class, we calculated a ubiquity factor u_c, which describes how often attribute class c is used in logs. We used the following formula:

$$u_c = n_c \cdot l_c / (L \cdot maxl_c) \tag{1}$$

where:

- n_c – number of occurrences of attribute class c in the selected papers,
- l_c – number of distinct log types in which attribute class c is reported in the selected papers,
- L – total number of log types identified in the selected papers,
- $maxl_c$ – maximum number of attribute occurrences over all attributes identified in the selected papers.

REFERENCES

[1] D. Yuan, S. Park, and Y. Zhou, "Characterizing logging practices in open-source software," in *Proc. 34th Int. Conf. Softw. Eng. (ICSE)*, Jun. 2012, pp. 102–112.
[2] B. Chen and Z. M. Jiang, "Characterizing logging practices in java-based open source software projects—A replication study in apache software foundation," *Empirical Softw. Eng.*, vol. 22, no. 1, pp. 330–374, Feb. 2017.
[3] *Splunk: The Data-to-Everything Platform*. Accessed: Sep. 5, 2021. [Online]. Available: https://www.splunk.com/
[4] *Logstash: Collect, Parse, Transform Logs|Elastic*. Accessed: Sep. 5, 2021. [Online]. Available: https://www.elastic.co/logstash/
[5] B. Kitchenham, "Procedures for performing systematic reviews," Dept. Comput. Sci., Keele Univ., Keele, U.K., Tech. Rep. 0400011T.1, 2004, pp. 1–26, vol. 33.
[6] B. Kitchenham and S. Charters, "Guidelines for performing systematic literature reviews in software engineering (version 2.3)," Keele Univ. Durham Univ., Keele, U.K., Tech. Rep. EBSE-2007-01, 2007.
[7] M. Schotten, M. El Aisati, W. J. N. Meester, S. Steiginga, and C. A. Ross, "A brief history of scopus: The world's largest abstract and citation database of scientific literature," in *Research Analytics*. Berlin, Germany: Auerbach, 2017, pp. 31–58, doi: 10.1201/9781315155890.
[8] P. Khandait, N. Tiwari, and N. Hubballi, "Who is trying to compromise your SSH server ? An analysis of authentication logs and detection of bruteforce attacks," in *Proc. Adjunct Proc. Int. Conf. Distrib. Comput. Netw.*, Jan. 2021, pp. 127–132.
[9] J. Ko and M. Comuzzi, "Detecting anomalies in business process event logs using statistical leverage," *Inf. Sci.*, vol. 549, pp. 53–67, Dec. 2021.
[10] Z. Zhao, C. Xu, and B. Li, "A LSTM-based anomaly detection model for log analysis," *J. Signal Process. Syst.*, vol. 93, no. 7, pp. 745–751, Jul. 2021.
[11] M. Cinque, R. Della Corte, V. Moscato, and G. Sperl, "A graph-based approach to detect unexplained sequences in a log," *Expert Syst. Appl.*, vol. 171, Dec. 2020, Art. no. 114556.
[12] B. Feng, C. Liu, and J. Yu, "A novel semantic user operation restoration from massive web URL log," in *Proc. IEEE 6th Int. Conf. Cloud Comput. Big Data Anal.*, Apr. 2021, pp. 261–266.
[13] M. Fuller, E. Brighton, M. Schiewe, D. Das, T. Cerny, and P. Tisnovsky, "Automated error log resolution: A case study," in *Proc. 36th Annu. ACM Symp. Appl. Comput.*, Mar. 2021, pp. 1298–1304.
[14] Y. Xie, K. Yang, and P. Luo, "LogM: Log Analysis for Multiple Components of Hadoop Platform," *IEEE Access*, vol. 9, pp. 73522–73532, 2021.
[15] W. Meng, Y. Liu, S. Zhang, F. Zaiter, Y. Zhang, Y. Huang, Z. Yu, Y. Zhang, L. Song, M. Zhang, and D. Pei, "LogClass: Anomalous log identification and classification with partial labels," *IEEE Trans. Netw. Service Manage.*, vol. 18, no. 2, pp. 1870–1884, Jun. 2021.
[16] V. Ramakrishnan and C. Palanisamy, "BIBSQLQC: Brown infomax boosted SQL query clustering algorithm to detect anti-patterns in the query log," *TURKISH J. Electr. Eng. Comput. Sci.*, vol. 28, no. 4, pp. 2200–2212, Jul. 2020.
[17] 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 *Proc. ACM/IEEE 42nd Int. Conf. Softw. Eng., Softw. Eng. Pract.*, Jun. 2020, pp. 71–80.
[18] B. Wang, S. Ying, G. Cheng, and Y. Li, "A log-based anomaly detection method with the NW ensemble rules," in *Proc. IEEE 20th Int. Conf. Softw. Qual., Rel. Secur. (QRS)*, Dec. 2020, pp. 72–82.
[19] S. Kim, A. Sim, K. Wu, S. Byna, Y. Son, and H. Eom, "Towards HPC I/O performance prediction through large-scale log analysis," in *Proc. 29th Int. Symp. High-Perform. Parallel Distrib. Comput.*, Jun. 2020, pp. 77–88.
[20] A. Pecchia, I. Weber, M. Cinque, and Y. Ma, "Discovering process models for the analysis of application failures under uncertainty of event logs," *Knowledge-Based Syst.*, vol. 189, Feb. 2020, Art. no. 105054.
[21] Y. Tao, S. Guo, C. Shi, and D. Chu, "User behavior analysis by cross-domain log data fusion," *IEEE Access*, vol. 8, pp. 400–406, 2020.
[22] J. P. Poh, J. Y. C. Lee, K. X. Tan, and E. Tan, "Physical access log analysis: An unsupervised clustering approach for anomaly detection," in *Proc. 3rd Int. Conf. Data Sci. Inf. Technol.*, Jul. 2020, pp. 12–18.
[23] T. Nishiyama, A. Kumagai, K. Kamiya, and K. Takahashi, "SILU: Strategy involving large-scale unlabeled logs for improving malware detector," in *Proc. IEEE Symp. Comput. Commun. (ISCC)*, Jul. 2020, pp. 1–7.
[24] M. Cinque, C. Esposito, and A. Pecchia, "Security log analysis in critical industrial systems exploiting game theoretic feature selection and evidence combination," *IEEE Trans. Ind. Informat.*, vol. 16, no. 6, pp. 3871–3880, Jun. 2020.
[25] Y. Pei and K. Oida, "Tracing website attackers by analyzing onion routers' log files," *IEEE Access*, vol. 8, pp. 133190–133203, 2020.
[26] C. M. Rosenberg and L. Moonen, "Spectrum-based log diagnosis," in *Proc. 14th ACM / IEEE Int. Symp. Empirical Softw. Eng. Meas. (ESEM)*, Oct. 2020, pp. 1–12.
[27] K. Yin, M. Yan, L. Xu, Z. Xu, Z. Li, D. Yang, and X. Zhang, "Improving log-based anomaly detection with component-aware analysis," in *Proc. IEEE Int. Conf. Softw. Maintenance Evol. (ICSME)*, Sep. 2020, pp. 667–671.

[28] Y. Ren, Z. Gu, Z. Wang, Z. Tian, C. Liu, H. Lu, and X. Du, "System log detection model based on conformal prediction," *Electronics*, vol. 9, no. 2, pp. 1–12, 2020.

[29] J. Chen, W. Shang, A. E. Hassan, Y. Wang, and J. Lin, "An experience report of generating load tests using log-recovered workloads at varying granularities of user behaviour," in *Proc. 34th IEEE/ACM Int. Conf. Automated Softw. Eng. (ASE)*, Nov. 2019, pp. 669–681.

[30] E. Chuah, A. Jhumka, S. Alt, D. Balouek-Thomert, J. C. Browne, and M. Parashar, "Towards comprehensive dependability-driven resource use and message log-analysis for HPC systems diagnosis," *J. Parallel Distrib. Comput.*, vol. 132, pp. 95–112, Dec. 2019.

[31] S. Lu, X. Wei, B. Rao, B. Tak, L. Wang, and L. Wang, "LADRA: Log-based abnormal task detection and root-cause analysis in big data processing with Spark," *Futur. Gener. Comput. Syst.*, vol. 95, pp. 392–403, Oct. 2019.

[32] P. Wu, Z. Lu, Q. Zhou, Z. Lei, X. Li, and M. Qiu, "Bigdata logs analysis based on seq2seq networks for cognitive Internet of Things," *Futur. Gener. Comput. Syst.*, vol. 90, pp. 477–488, Jan. 2019.

[33] Y. Yuan, W. Shi, B. Liang, and B. Qin, "An approach to cloud execution failure diagnosis based on exception logs in OpenStack," in *Proc. IEEE 12th Int. Conf. Cloud Comput. (CLOUD)*, Jul. 2019, pp. 124–131.

[34] Y. Yuan, H. Anu, W. Shi, B. Liang, and B. Qin, "Learning-based anomaly cause tracing with synthetic analysis of logs from multiple cloud service components," in *Proc. IEEE 43rd Annu. Comput. Softw. Appl. Conf. (COMPSAC)*, Jul. 2019, pp. 66–71.

[35] D. Barua, N. T. Rumpa, S. Hossen, and M. M. Ali, "Ontology based log analysis of web servers using process mining techniques," in *Proc. 10th Int. Conf. Electr. Comput. Eng. (ICECE)*, Dec. 2018, pp. 341–344.

[36] M. Astekin, H. Zengin, and H. Sozer, "Evaluation of distributed machine learning algorithms for anomaly detection from large-scale system logs: A case study," in *Proc. IEEE Int. Conf. Big Data (Big Data)*, Dec. 2018, pp. 2071–2077.

[37] H. L. Ngoc, T. Cong Hung, N. D. Huy, and N. Thi Thanh Hang, "Early phase warning solution about system security based on log analysis," in *Proc. 6th NAFOSTED Conf. Inf. Comput. Sci. (NICS)*, Dec. 2019, pp. 398–403.

[38] K. Dan, N. Kitagawa, S. Sakuraba, and N. Yamai, "Spam domain detection method using active DNS data and E-Mail reception log," in *Proc. IEEE 43rd Annu. Comput. Softw. Appl. Conf. (COMPSAC)*, Jul. 2019, pp. 896–899.

[39] D. Kim, D. Shin, D. Shin, and Y.-H. Kim, "Attack detection application with attack tree for mobile system using log analysis," *Mobile Netw. Appl.*, vol. 24, no. 1, pp. 184–192, Feb. 2019.

[40] D. Kim, Y. H. Kim, D. Shin, and D. Shin, "Fast attack detection system using log analysis and attack tree generation," *Cluster Comput.*, vol. 22, no. s1, pp. 1827–1835, 2019.

[41] P. Sai Charan, "Abnormal user pattern detection Using semi-structured server log file analysis," in *Smart Intelligent Computing and Applications*. Singapore: Springer, 2018, pp. 97–105.

[42] M. Ait El Hadj, A. Khoumsi, Y. Benkaouz, and M. Erradi, "Efficient security policy management using suspicious rules through access log analysis," in *Networked Systems* (Lecture Notes in Computer Science), vol. 11704. Cham, Switzerland: Springer, Jan. 2020, pp. 250–266, 2019.

[43] Y. Tao, Y. Wang, C. Shi, X. Wang, C. Xu, and Z. Xu, "Cross-domain user profile construction by log analysis," in *Proc. IEEE 4th Int. Conf. Data Sci. Cyberspace (DSC)*, Jun. 2019, pp. 217–221.

[44] X. Xia, W. Zhang, and J. Jiang, "Ensemble methods for anomaly detection based on system log," in *Proc. IEEE 24th Pacific Rim Int. Symp. Dependable Comput. (PRDC)*, Dec. 2019, pp. 93–94.

[45] M. Astekin, S. Ozcan, and H. Sozer, "Incremental analysis of large-scale system logs for anomaly detection," in *Proc. IEEE Int. Conf. Big Data (Big Data)*, Dec. 2019, pp. 2119–2127.

[46] X. Zhang, Y. Xu, Q. Lin, and B. Qiao, "Robust log-based anomaly detection on unstable log data," in *Proc. 27th ACM Joint Meeting Eur. Softw. Eng. Conf. Symp. Found. Softw. Eng.*, Aug. 2019, pp. 807–817.

[47] X. Wang, D. Wang, Y. Zhang, L. Jin, and M. Song, "Unsupervised learning for log data analysis based on behavior and attribute features," in *Proc. Int. Conf. Artif. Intell. Comput. Sci.*, Jul. 2019, pp. 510–518.

[48] A. Patil, A. Wadekar, T. Gupta, R. Vijan, and F. Kazi, "Explainable LSTM model for anomaly detection in HDFS log file using layerwise relevance propagation," in *Proc. IEEE Bombay Sect. Signature Conf. (IBSSC)*, Jul. 2019, pp. 1–6.

[49] A. Wadekar, T. Gupta, R. Vijan, and F. Kazi, "Hybrid CAE-VAE for unsupervised anomaly detection in log file systems," in *Proc. 10th Int. Conf. Comput., Commun. Netw. Technol. (ICCCNT)*, Jul. 2019, pp. 1–7.

[50] M. Astekin, H. Zengin, and H. Sözer, "DILAF: A framework for distributed analysis of large-scale system logs for anomaly detection," *Softw., Pract. Exper.*, vol. 49, no. 2, pp. 153–170, Feb. 2019.

[51] Y. Cui, Y. Sun, J. Hu, and G. Sheng, "A convolutional auto-encoder method for anomaly detection on system logs," in *Proc. IEEE Int. Conf. Syst., Man, Cybern. (SMC)*, Oct. 2018, pp. 3057–3062.

[52] P. Robberechts, M. Bosteels, J. Davis, and W. Meert, "Query log analysis: Detecting anomalies in DNS traffic at a TLD resolver," *Commun. Comput. Inf. Sci.*, vol. 967, pp. 55–67, Sep. 2019.

[53] K. Otomo, S. Kobayashi, K. Fukuda, and H. Esaki, "Latent variable based anomaly detection in network system logs," *IEICE Trans. Inf. Syst.*, vol. E102D, no. 9, pp. 1644–1652, 2019.

[54] N. Arzamasova, M. Schaler, and K. Bohm, "Cleaning antipatterns in an SQL query log," in *Proc. IEEE 34th Int. Conf. Data Eng. (ICDE)*, Apr. 2018, pp. 1751–1752.

[55] C. M. Rosenberg and L. Moonen, "Improving problem identification via automated log clustering using dimensionality reduction," in *Proc. 12th ACM/IEEE Int. Symp. Empirical Softw. Eng. Meas.*, Oct. 2018, pp. 1–10.

[56] Z. Li, M. Davidson, S. Fu, S. Blanchard, and M. Lang, "Converting unstructured system logs into structured event list for anomaly detection," in *Proc. 13th Int. Conf. Availability, Rel. Secur.*, Aug. 2018, pp. 1–10.

[57] R. Chen, Q. Gao, W. Ji, F. Long, and Q. Ling, "Network log analysis based on the topic word mover's distance," in *Proc. Chin. Control Decis. Conf. (CCDC)*, Jun. 2018, pp. 4082–4086.

[58] S. Kim and D. Kim, "Analyzing mobile application logs using process mining techniques: An application to online bookstores," *ICIC Exp. Lett. Part B Appl.*, vol. 9, no. 6, pp. 607–614, 2018.

[59] S. K. Tun, "A web application path analysis through server logs," in *Proc. 10th Int. Jt. Conf. Knowl. Discov. Knowl. Eng. Knowl. Manag.*, vol. 1, 2018, pp. 427–430.

[60] M. Kaur, F. D. Salim, Y. Ren, J. Chan, M. Tomko, and M. Sanderson, "Shopping intent recognition and location prediction from cyber-physical activities via Wi-Fi logs," in *Proc. 5th Conf. Syst. Built Environ.*, Nov. 2018, pp. 130–139.

[61] M. Farshchi, J.-G. Schneider, I. Weber, and J. Grundy, "Metric selection and anomaly detection for cloud operations using log and metric correlation analysis," *J. Syst. Softw.*, vol. 137, pp. 531–549, Mar. 2017.

[62] S. He, Q. Lin, J.-G. Lou, H. Zhang, M. R. Lyu, and D. Zhang, "Identifying impactful service system problems via log analysis," in *Proc. 26th ACM Joint Meeting Eur. Softw. Eng. Conf. Symp. Found. Softw. Eng.*, Oct. 2018, pp. 60–70.

[63] E. Chuah, A. Jhumka, S. Alt, T. Damoulas, N. Gurumdimma, M.-C. Sawley, W. L. Barth, T. Minyard, and J. C. Browne, "Enabling dependability-driven resource use and message log-analysis for cluster system diagnosis," in *Proc. IEEE 24th Int. Conf. High Perform. Comput. (HiPC)*, Dec. 2017, pp. 317–327.

[64] S. Khan and S. Parkinson, "Eliciting and utilising knowledge for security event log analysis: An association rule mining and automated planning approach," *Expert Syst. Appl.*, vol. 113, no. July, pp. 116–127, 2018.

[65] M. Leemans, W. M. P. van der Aalst, and M. G. J. van den Brand, "Recursion aware modeling and discovery for hierarchical software event log analysis," in *Proc. IEEE 25th Int. Conf. Softw. Anal., Evol. Reeng. (SANER)*, Mar. 2018, pp. 185–196.

[66] Y. Djenouri, A. Belhadi, and P. Fournier-Viger, "Extracting useful knowledge from event logs: A frequent itemset mining approach," *Knowl.-Based Syst.*, vol. 139, pp. 132–148, Oct. 2018.

[67] E. U. Aktas, M. C. Calpur, and U. U. Yildirim, "Inferring dependencies among web services with predictive and statistical analysis of system logs," in *Proc. CEUR Workshop*, vol. 2291, Dec. 2018, pp. 235–244.

[68] C. Qiao and X. Hu, "Discovering Student behavior patterns from event logs: Preliminary results on a novel probabilistic latent variable model," in *Proc. IEEE 18th Int. Conf. Adv. Learn. Technol. (ICALT)*, Jul. 2018, pp. 207–211.

[69] R. C. Raga and J. D. Raga, "A comparison of college faculty and student class activity in an online learning environment using course log data," in *Proc. IEEE SmartWorld Ubiquitous Intell. Comput. Adv. Trust. Comput. Scalable*, Dec. 2018, pp. 1–6.
[70] E. A. Gamie, M. S. A. El-Seoud, M. A. Salama, and W. Hussein, "Pedagogical and elearning logs analyses to enhance students' performance," in *Proc. 7th Int. Conf. Softw. Inf. Eng.*, 2018, pp. 116–120.
[71] T. Shibahara, K. Yamanishi, and Y. Takata, "Event de-noising convolutional neural network for detecting malicious URL sequences from proxy Logs," *IEICE Trans. Fundam. Electron. Commun. Comput. Sci.*, vol. E101A, no. 12, pp. 2149–2161, 2018.
[72] D. Zhang, Y. Zheng, Y. Wen, Y. Xu, J. Wang, Y. Yu, and D. Meng, "Role-based log analysis applying deep learning for insider threat detection," in *Proc. 1st Workshop Secur.-Oriented Designs Comput. Archit. Processors*, Jan. 2018, pp. 18–20.
[73] Q. Cao, Y. Qiao, and Z. Lyu, "Machine learning to detect anomalies in web log analysis," in *Proc. 3rd IEEE Int. Conf. Comput. Commun. (ICCC)*, Dec. 2017, pp. 519–523.
[74] M. Bas Seyyar, F. Ö. Çatak, and E. Gül, "Detection of attack-targeted scans from the apache HTTP server access logs," *Appl. Comput. Informat.*, vol. 14, no. 1, pp. 28–36, Jan. 2018.
[75] J. Navarro, V. Legrand, S. Lagraa, and J. François, "HuMa: A multi-layer framework for threat analysis in a heterogeneous log environment," in *Proc. Int. Symp. Found. Pract. Secur.*, vol. 10723, Feb. 2018, pp. 144–159.
[76] A. Brown, A. Tuor, B. Hutchinson, and N. Nichols, "Recurrent neural network attention mechanisms for interpretable system log anomaly detection," in *Proc. Workshop Mach. Learn. Comput. Syst.*, 2018, pp. 1–8.
[77] S. H. Sim, H. Bae, Y. Choi, and L. Liu, "Statistical verification of process model conformance to execution log considering model abstraction," *Int. J. Coop. Inf. Syst.*, vol. 27, no. 2, 2018, Art. no. 1850002.
[78] M. Wang, L. Xu, and L. Guo, "Anomaly detection of system logs based on natural language processing and deep learning," in *Proc. 4th Int. Conf. Frontiers Signal Process. (ICFSP)*, Sep. 2018, pp. 140–144.
[79] S. Lu, X. Wei, Y. Li, and L. Wang, "Detecting anomaly in big data system logs using convolutional neural network," in *Proc. IEEE 16th Int. Conf. Dependable, Autonomic Secure Comput.*, Aug. 2018, pp. 159–165.
[80] Y. Liu, J. Lv, S. Ma, and W. Yao, "The runtime system problem identification method based on log analysis," in *Proc. 27th Int. Conf. Comput. Commun. Netw. (ICCCN)*, Jul. 2018, pp. 1–7.
[81] L. Bao, Q. Li, P. Lu, J. Lu, T. Ruan, and K. Zhang, "Execution anomaly detection in large-scale systems through console log analysis," *J. Syst. Softw.*, vol. 143, no. May, pp. 172–186, 2018.
[82] K. Otomo, S. Kobayashi, K. Fukuda, and H. Esaki, "Finding anomalies in network system logs with latent variables," in *Proc. Workshop Big Data Anal. Mach. Learn. Data Commun. Netw.*, 2018, pp. 8–14.
[83] B. Debnath, M. Solaimani, M. A. G. Gulzar, N. Arora, C. Lumezanu, J. Xu, B. Zong, H. Zhang, G. Jiang, and L. Khan, "LogLens: A real-time log analysis system," in *Proc. IEEE 38th Int. Conf. Distrib. Comput. Syst. (ICDCS)*, Jul. 2018, pp. 1052–1062.
[84] R. Rastogi, S. Nahata, P. Ghuli, D. Pratiba, and G. Shobha, "Anomaly detection in log records," *Indones. J. Electr. Eng. Comput. Sci.*, vol. 10, no. 1, pp. 343–347, 2018.
[85] M. Dietz and G. Pernul, "Big log data stream processing: Adapting an anomaly detection technique," in *Proc. Int. Conf. Database Expert Syst. Appl.*, 2018, pp. 159–166.
[86] M. Wurzenberger, F. Skopik, G. Settanni, and R. Fiedler, "AECID: A self-learning anomaly detection approach based on light-weight log parser models," in *Proc. 4th Int. Conf. Inf. Syst. Secur. Privacy*, 2018, pp. 386–397.
[87] A. Ekelhart, E. Kiesling, and K. Kurniawan, "Taming the logs—Vocabularies for semantic security analysis," *Proc. Comput. Sci.*, vol. 137, pp. 109–119, Jan. 2018.
[88] Z. Liu, T. Qin, X. Guan, H. Jiang, and C. Wang, "An integrated method for anomaly detection from massive system logs," *IEEE Access*, vol. 6, pp. 30602–30611, 2018.
[89] D. Gadler, M. Mairegger, A. Janes, and B. Russo, "Mining logs to model the use of a system," in *Proc. ACM/IEEE Int. Symp. Empirical Softw. Eng. Meas. (ESEM)*, Nov. 2017, pp. 334–343.
[90] J. Xu, P. Chen, L. Yang, F. Meng, and P. Wang, "LogDC: Problem diagnosis for declartively-deployed cloud applications with log," in *Proc. IEEE 14th Int. Conf. E-Bus. Eng. (ICEBE)*, Nov. 2017, pp. 282–287.
[91] S. Lu, B. Rao, X. Wei, B. Tak, L. Wang, and L. Wang, "Log-based abnormal task detection and root cause analysis for spark," in *Proc. IEEE Int. Conf. Web Services (ICWS)*, Jun. 2017, pp. 389–396.
[92] G. Qi, W. T. Tsai, W. Li, Z. Zhu, and Y. Luo, "A cloud-based triage log analysis and recovery framework," *Simul. Model. Pract. Theory*, vol. 77, pp. 292–316, Aug. 2020.
[93] X. Tian, H. Li, and F. Liu, "Web service reliability test method based on log analysis," in *Proc. IEEE Int. Conf. Softw., Rel. Secur. Companion (QRS-C)*, Jul. 2017, pp. 195–199.
[94] J. Breier and J. Branišová, "A dynamic rule creation based anomaly detection method for identifying security breaches in log records," *Wireless Pers. Commun.*, vol. 94, no. 3, pp. 497–511, Jun. 2017.
[95] T. Ayaki, H. Yanagimoto, and M. Yoshioka, "Recommendation from access logs with ensemble learning," *Artif. Life Robot.*, vol. 22, no. 2, pp. 163–167, Jun. 2017.
[96] C. López, R. Heinsen, and E.-N. Huh, "Improving availability applying intelligent replication in federated cloud storage based on log analysis," in *Proc. Int. Conf. Mach. Learn. Soft Comput.*, Jan. 2017, pp. 148–153.
[97] K. Ma, R. Jiang, M. Dong, Y. Jia, and A. Li, "Neural network based web log analysis for web intrusion detection," in *Proc. Secur., Privacy, Anonymity Comput., Commun., Storage Lect. Notes Comput. Sci.*, 2017, pp. 194–204.
[98] H. Labbaci, B. Medjahed, and Y. Aklouf, "Learning interactions from web service logs," in *Proc. Int. Conf. Database Expert Syst. Appl.*, vol. 10439, Aug. 2017, pp. 275–289.
[99] N. M. F. Qureshi, D. R. Shin, I. F. Siddiqui, and A. Abbas, "HSLA: Heterogeneous storage-tier log analyzer over Hadoop," *J. Theor. Appl. Inf. Technol.*, vol. 95, no. 10, pp. 2290–2296, 2017.
[100] W.-T. Lin and J.-Y. Pan, "Mobile malware detection in sandbox with live event feeding and log pattern analysis," in *Proc. 18th Asia–Pacific Netw. Oper. Manage. Symp. (APNOMS)*, Oct. 2016, pp. 1–7.
[101] J. He, P. Qvarfordt, M. Halvey, and G. Golovchinsky, "Beyond actions: Exploring the discovery of tactics from user logs," *Inf. Process. Manage.*, vol. 52, no. 6, pp. 1200–1226, Nov. 2016.
[102] D. Zou, H. Qin, and H. Jin, "UiLog: Improving log-based fault diagnosis by log analysis," *J. Comput. Sci. Technol.*, vol. 31, no. 5, pp. 1038–1052, 2016, doi: 10.1007/s11390-016-1678-7.
[103] M. Moh, S. Pininti, S. Doddapaneni, and T.-S. Moh, "Detecting web attacks using multi-stage log analysis," in *Proc. IEEE 6th Int. Conf. Adv. Comput. (IACC)*, Feb. 2016, pp. 733–738.
[104] N. M. Zaki, A. Awad, and E. Ezat, "Extracting accurate performance indicators from execution logs using process models," in *Proc. IEEE/ACS 12th Int. Conf. Comput. Syst. Appl. (AICCSA)*, Nov. 2015, pp. 1–8.
[105] X. Yu, P. Joshi, J. Xu, G. Jin, H. Zhang, and G. Jiang, "CloudSeer: Workflow monitoring of cloud infrastructures via interleaved logs," in *Proc. Int. Conf. Archit. Support Program. Lang. Oper. Syst.*, vols. 2, 2016, pp. 489–502.
[106] S. Saito, K. Maruhashi, M. Takenaka, and S. Torii, "TOPASE: Detection and prevention of brute force attacks with disciplined IPs from IDS logs," *J. Inf. Process.*, vol. 24, no. 2, pp. 217–226, 2016.
[107] R. White, S. Wang, A. Pant, R. Harpaz, and P. Shukla, "Early identification of adverse drug reactions from search log data," *J. Biomed. Inform.*, vol. 59, pp. 42–48, Feb. 2016.
[108] A. Iswardani and I. Riadi, "Denial of service log analysis using density K-means method," *J. Theor. Appl. Inf. Technol.*, vol. 83, no. 2, pp. 299–302, 2016.
[109] M. Farshchi, J.-G. Schneider, I. Weber, and J. Grundy, "Experience report: Anomaly detection of cloud application operations using log and cloud metric correlation analysis," in *Proc. IEEE 26th Int. Symp. Softw. Rel. Eng. (ISSRE)*, Nov. 2015, pp. 24–34.
[110] H. Sun, J. Sun, and H. Chen, "Mining frequent attack sequence in web logs," in *Green, Pervasive, and Cloud Computing* (Lecture Notes in Computer Science), vol. 9663. Cham, Switzerland: Springer, 2016, pp. 243–260.
[111] A. I. Hajamydeen, N. I. Udzir, R. Mahmod, and A. A. Abdul Ghani, "An unsupervised heterogeneous log-based framework for anomaly detection," *TURKISH J. Electr. Eng. Comput. Sci.*, vol. 24, pp. 1117–1134, Oct. 2016.
[112] K.-S. Jeon, S.-J. Park, S.-H. Chun, and J.-B. Kim, "A study on the big data log analysis for security," *Int. J. Secur. Appl.*, vol. 10, no. 1, pp. 13–20, Jan. 2016.
[113] S.-L. Chuang and L.-F. Chien, "Enriching web taxonomies through subject categorization of query terms from search engine logs," *Decis. Support Syst.*, vol. 35, no. 1, pp. 113–127, Apr. 2003.
[114] R. Baeza-Yates and A. Tiberi, "Extracting semantic relations from query logs," in *Proc. 13th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2007, pp. 76–85.

[115] W. Xu, L. Huang, A. Fox, D. Patterson, and M. I. Jordan, "Detecting large-scale system problems by mining console logs," in *Proc. ACM SIGOPS 22nd Symp. Oper. Syst. Princ.*, 2009, pp. 117–132.

[116] A. Andrejko and M. Barla, "Rule-based user characteristics acquisition from logs with semantics for personalized web-based systems," in *Proc. CEUR Workshop*, May 2014, vol. 252, pp. 103–110.

[117] W. Gaaloul, K. Gaaloul, S. Bhiri, A. Haller, and M. Hauswirth, "Log-based transactional workflow mining," *Distrib. Parallel Databases*, vol. 25, no. 3, pp. 193–240, Jun. 2009.

[118] J.-G. Lou, Q. Fu, Y. Wang, and J. Li, "Mining dependency in distributed systems through unstructured logs analysis," *ACM SIGOPS Oper. Syst. Rev.*, vol. 44, no. 1, pp. 91–96, Mar. 2010.

[119] Y. Hu, Y. Qian, H. Li, D. Jiang, J. Pei, and Q. Zheng, "Mining query subtopics from search log data," in *Proc. 35th Int. ACM SIGIR Conf. Res. Develop. Inf. Retr.*, 2012, pp. 305–314.

[120] D. Hadžiosmanovic, D. Bolzoni, and P. H. Hartel, "A log mining approach for process monitoring in SCADA," *Int. J. Inf. Secur.*, vol. 11, no. 4, pp. 231–251, Aug. 2012.

[121] A. R. Kang, J. Woo, J. Park, and H. K. Kim, "Online game bot detection based on party-play log analysis," *Comput. Math. Appl.*, vol. 65, no. 9, pp. 1384–1395, May 2013.

[122] I. Beschastnikh, Y. Brun, M. D. Ernst, and A. Krishnamurthy, "Inferring models of concurrent systems from logs of their behavior with CSight," in *Proc. 36th Int. Conf. Softw. Eng.*, 2014, pp. 468–479.

[123] A. Ambre and N. Shekokar, "Insider threat detection using log analysis and event correlation," *Proc. Comput. Sci.*, vol. 45, pp. 436–445, Oct. 2015.

[124] A. Fourney, R. Mann, and M. Terry, "Characterizing the usability of interactive applications through query log analysis," in *Proc. SIGCHI Conf. Hum. Factors Comput. Syst.*, May 2011, pp. 1817–1826.

[125] A. R. Arasteh, M. Debbabi, A. Sakha, and M. Saleh, "Analyzing multiple logs for forensic evidence," *Digit. Investig.*, vol. 4, pp. 82–91, Oct. 2007.

[126] N. N. Vo, S. Liu, X. Li, and G. Xu, "Leveraging unstructured call log data for customer churn prediction," *Knowl.-Based Syst.*, vol. 212, Jun. 2021, Art. no. 106586.

[127] A. Mudholkar, V. Mokhashi, D. Nayak, V. Annavarjula, and M. B. Jayaraman, "Analysis of automated log template generation methodologies," in *Proc. Adv. Intell. Syst. Comput. Adv. Artif. Intell. Data Eng.*, 2020, pp. 571–588.

[128] R. K. Fischer, A. Iglesias, A. L. Daugherty, and Z. Jiang, "A transaction log analysis of EBSCO Discovery Service using Google Analytics: The methodology," *Library Tech*, vol. 39, no. 1, pp. 249–262, Dec. 2020.

[129] J. Cándido, M. Aniche, and A. Van Deursen, "Log-based software monitoring: A systematic mapping study," *PeerJ Comput. Sci.*, vol. 7, pp. 1–38, Oct. 2021.

[130] S. Locke, H. Li, T. H. P. Chen, W. Shang, and W. Liu, "LogAssist: Assisting log analysis through log summarization," *IEEE Trans. Softw. Eng.*, early access, May 26, 2021, doi: 10.1109/TSE.2021.3083715.

[131] F. Zhou and H. Qu, "A GMM-based anomaly IP detection model from security logs," in *Proc. Int. Conf. Smart Comput. Commun.*, 2021, pp. 97–105.

[132] Z. Bin, S. Shuai, G. Zhi-chun, and H. Jian-feng, "Design and implementation of incremental data capturing in wireless network planning based on log mining," in *Proc. IEEE 5th Adv. Inf. Technol., Electron. Autom. Control Conf. (IAEAC)*, Mar. 2021, pp. 2757–2761.

[133] S. Kobayashi, Y. Yamashiro, K. Otomo, and K. Fukuda, "Amulog: A general log analysis framework for diverse template generation methods," in *Proc. 16th Int. Conf. Netw. Service Manage. (CNSM)*, Nov. 2020, pp. 1–5.

[134] J. Svacina, J. Raffety, C. Woodahl, B. Stone, T. Cerny, M. Bures, D. Shin, K. Frajtak, and P. Tisnovsky, "On vulnerability and security log analysis: A systematic literature review on recent trends," in *Proc. Int. Conf. Res. Adapt. Convergent Syst.*, Oct. 2020, pp. 175–180.

[135] V. Bushong, R. Sanders, J. Curtis, M. Du, T. Cerny, K. Frajtak, M. Bures, P. Tisnovsky, and D. Shin, "On matching log analysis to source code," in *Proc. Int. Conf. Res. Adapt. Convergent Syst.*, Oct. 2020, pp. 181–187.

[136] D. Das, M. Schiewe, E. Brighton, M. Fuller, T. Cerny, M. Bures, K. Frajtak, D. Shin, and P. Tisnovsky, "Failure prediction by utilizing log analysis: A systematic mapping study," in *Proc. Int. Conf. Res. Adapt. Convergent Syst.*, Oct. 2020, pp. 188–195.

[137] M. Wurzenberger, G. Höld, M. Landauer, F. Skopik, and W. Kastner, "Creating character-based templates for log data to enable security event classification," in *Proc. 15th ACM Asia Conf. Comput. Commun. Secur.*, Oct. 2020, pp. 141–152.

[138] S. Halle, "Explainable queries over event logs," in *Proc. IEEE 24th Int. Enterprise Distrib. Object Comput. Conf. (EDOC)*, Oct. 2020, pp. 171–180.

[139] W. Meng, Y. Liu, Y. Huang, S. Zhang, F. Zaiter, B. Chen, and D. Pei, "A semantic-aware representation framework for online log analysis," in *Proc. 29th Int. Conf. Comput. Commun. Netw. (ICCCN)*, Aug. 2020, pp. 1–7.

[140] W. Meng, Y. Liu, F. Zaiter, S. Zhang, Y. Chen, Y. Zhang, Y. Zhu, E. Wang, R. Zhang, S. Tao, D. Yang, R. Zhou, and D. Pei, "LogParse: Making log parsing adaptive through word classification," in *Proc. 29th Int. Conf. Comput. Commun. Netw. (ICCCN)*, Aug. 2020, pp. 1–9.

[141] C.-K. Tsung, C.-T. Yang, and S.-W. Yang, "Visualizing potential transportation demand from ETC log analysis using ELK stack," *IEEE Internet Things J.*, vol. 7, no. 7, pp. 6623–6633, Jul. 2020.

[142] B. Zhang, Z. Xi, T. Zhang, Y. Ma, Z. Shao, and H. Li, "Dimensionality reduction of massive I/O log data flow in power system," in *Proc. Int. Conf. Commun., Inf. Syst. Comput. Eng. (CISCE)*, Jul. 2020, pp. 198–202.

[143] C. E. Brandt, A. Panichella, A. Zaidman, and M. Beller, "LogChunks: A data set for build log analysis," in *Proc. 17th Int. Conf. Mining Softw. Repositories*, Jun. 2020, pp. 583–587.

[144] D. El-Masri, F. Petrillo, A. Hamou-Lhadj, and A. Bouziane, "A systematic literature review on automated log abstraction techniques," *Inf. Softw. Technol.*, vol. 122, Mar. 2020, Art. no. 106276.

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

[146] W. Yuan, S. Lu, H. Sun, and X. Liu. (Nov. 20, 2019). *How Are Distributed Bugs Diagnosed and Fixed Through System Logs*. Accessed: Sep. 5, 2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0950584919302496

[147] V. Carchiolo, A. Longheu, G. Saccullo, S. Sau, and R. Sortino, "ICs manufacturing workflow assessment via multiple logs analysis," in *Proc. 22nd Int. Conf. Enterprise Inf. Syst.*, 2020, pp. 801–809.

[148] D. L. Ivanova, S. V. Kutepov, and A. A. Dyumin, "Fibre channel switch port state machine analysis based on the port log dump," in *Proc. IEEE Conf. Russian Young Researchers Electr. Electron. Eng. (EIConRus)*, Jan. 2020, pp. 24–27.

[149] T. Kulahcioglu, D. Fradkin, A. Parlak, and A. Belkov, "LogVis: Graph-assisted visual analysis of event logs from industrial equipment," in *Proc. CEUR Workshop*, 2020, p. 61.

[150] R. Israel-Fishelson, A. Hershkovitz, A. Eguíluz, P. Garaizar, and M. Guenaga, "A log-based analysis of the associations between creativity and computational thinking," *J. Educ. Comput. Res.*, vol. 59, no. 5, pp. 926–959, Sep. 2021.

[151] S. Isaev, D. Kononov, and A. Malyshev, "Analysis of internet service log data to assess the level of cyber-threats in the corporate network," in *Proc. CEUR Workshop Proc*, 2020, p. 16.

[152] A. K. Meena and N. Hubballi, "NViZ: An interactive visualization of network security systems logs," in *Proc. Int. Conf. Commun. Syst. Netw. (COMSNETS)*, Jan. 2020, pp. 685–687.

[153] D. Hofer and A. Mohamed, "On applying graph database time models for security log analysis," *Proc. Int. Conf. Future Data Secur. Eng.* (Lecture Notes in Computer Science). Cham, Switzerland: Springer, 2020, pp. 87–107.

[154] E. Haihong, Y. Chen, M. Song, and M. Sun, "Distributed cloud monitoring platform based on log in-sight," in *Int. Conf. Cloud Comput.*, 2020, pp. 76–88.

[155] K. Nakamura, G. Kosugi, T. Sato, E. Morita, and Y. Hayashi, "Prototyping of log analysis infrastructure for the Subaru telescope based on the ALMA experience," *Observatory Oper., Strategies, Processes, Syst.*, vol. 1149, pp. 524–531, Dec. 2020.

[156] M. Kaepke and O. Zukunft, "A comparative evaluation of big data frameworks for graph processing," in *Proc. 4th Int. Conf. Big Data Innov. Appl.*, Aug. 2018, p. 57.

[157] A. Bonifati, W. Martens, and T. Timm, "An analytical study of large SPARQL query logs," *VLDB J.*, vol. 29, nos. 2–3, pp. 655–679, 2020.

[158] D. Obrâbski and J. Sosnowski, "Log based analysis of software application operation," *Adv. Intell. Syst. Comput.*, vol. 987, pp. 371–382, Oct. 2020.

[159] P. Prasad and S. Iyer, "Inferring students' tracing behaviors from interaction logs of a learning environment for software design comprehension," in *Proc. Koli Calling*, Nov. 2020, pp. 1–5.

[160] M. Naseer, W. Zhang, and W. Zhu, "A framework to strengthen up business interests in students by using matrix factorization on web log," in *Proc. Commun. Comput. Inf. Sci. Intell. Technol. Appl.*, 2020, pp. 322–332.
[161] T.-Y. Kim, J.-Y. Gang, and H.-J. Oh, "Spatial usage analysis based on user activity big data logs in library," *Library Tech.*, vol. 38, no. 3, pp. 678–698, Nov. 2019.
[162] J. Lian, "Implementation of computer network user behavior forensic analysis system based on speech data system log," *Int. J. Speech Technol.*, vol. 23, no. 3, pp. 559–567, Sep. 2020.
[163] T. Eljasik-Swoboda and W. Demuth, "Leveraging clustering and natural language processing to overcome variety issues in log management," in *Proc. 12th Int. Conf. Agents Artif. Intell.*, 2020, pp. 281–288.
[164] T. Schmidt, F. Hauer, and A. Pretschner, "Automated anomaly detection in CPS log files," *Proc. Int. Conf. Comput. Saf., Rel., Secur.*, 2020, pp. 179–194.
[165] A. Bonifati, W. Martens, and T. Timm, "Navigating the maze of Wikidata query logs," in *Proc. World Wide Web Conf.*, 2019, pp. 127–138.
[166] K. Matsumoto, T. Nakanishi, and T. Kitagawa, "Semantic-dependent access log analysis for predicting the demographic data," in *Proc. Conf. Inf. Modeling Knowl. Bases*, 2019, pp. 415–434.
[167] F. Shilpika, B. Lusch, M. Emani, V. Vishwanath, M. E. Papka, and K.-L. Ma, "MELA: A visual analytics tool for studying multifidelity HPC system logs," in *Proc. IEEE/ACM Ind./Univ. Joint Int. Workshop Data-Center Autom., Anal., Control (DAAC)*, Nov. 2019, pp. 13–18.
[168] L. Bao, N. Busany, D. Lo, and S. Maoz, "Statistical log differencing," in *Proc. 34th IEEE/ACM Int. Conf. Automated Softw. Eng. (ASE)*, Nov. 2019, pp. 851–862.
[169] U. Krishnan, B. Billerbeck, A. Moffat, and J. Zobel, "Abstraction of query auto completion logs for anonymity-preserving analysis," *Inf. Retr. J.*, vol. 22, no. 5, pp. 499–524, Oct. 2019.
[170] G. Li, P. Zhu, N. Cao, M. Wu, Z. Chen, G. Cao, H. Li, and C. Gong, "Improving the system log analysis with language model and semi-supervised classifier," *Multimedia Tools Appl.*, vol. 78, no. 15, pp. 21521–21535, Aug. 2019.
[171] F. Pompili, J. G. Conrad, and C. Kolbeck, "Exploiting search logs to aid in training and automating infrastructure for question answering in professional domains," in *Proc. 17th Int. Conf. Artif. Intell. Law*, Jun. 2019, pp. 93–102.
[172] M. Dunaev and K. Zaytsev, "Logs analysis to search for anomalies in the functioning of large technology platforms," *J. Theor. Appl. Inf. Technol.*, vol. 97, no. 11, pp. 3111–3123, 2019.
[173] I. Itkin, A. Gromova, A. Sitnikov, D. Legchikov, E. Tsymbalov, R. Yavorskiy, A. Novikov, and K. Rudakov, "User-assisted log analysis for quality control of distributed fintech applications," in *Proc. IEEE Int. Conf. Artif. Intell. Test. (AITest)*, Apr. 2019, pp. 45–51.
[174] M. Wurzenberger, M. Landauer, F. Skopik, and W. Kastner, "AECID-PG: A tree-based log parser generator to enable log analysis," in *Proc. Symp. Integr. Netw. Serv. Manag.*, 2019, pp. 7–12.
[175] A. Rafique, K. Ameen, and A. Arshad, "Use patterns of e-journals among the science community: A transaction log analysis," *Electron. Library*, vol. 37, no. 4, pp. 740–759, Sep. 2019.
[176] J. Zhu, S. He, J. Liu, P. He, Q. Xie, Z. Zheng, and M. R. Lyu, "Tools and benchmarks for automated log parsing," in *Proc. IEEE/ACM 41st Int. Conf. Softw. Eng., Softw. Eng. Pract. (ICSE-SEIP)*, May 2019, pp. 121–130.
[177] J. Duan, C. Yang, and J. He, "A ROP optimization approach based on well log data analysis using deep learning network and PSO," in *Proc. IEEE Int. Conf. Intell. Appl. Syst. Eng. (ICIASE)*, Apr. 2019, pp. 86–88.
[178] C. Wickramage, C. Fidge, C. Ouyang, and T. Sahama, "Generating log requirements for checking conformance against healthcare standards using workflow modelling," in *Proc. Australas. Comput. Sci. Week Multiconference*, Jan. 2019, pp. 1–10.
[179] J. Liu, Z. Hou, and Y. Li, "Lopper: An efficient method for online log pattern mining based on hybrid clustering tree," in *Proc. Int. Conf. Database Expert Syst. Appl.*, 2019, pp. 63–78.
[180] A. Luoto, "Log analysis of 360-degree video users via MQTT," in *Proc. 2nd Int. Conf. Geoinformatics Data Anal.*, Mar. 2019, pp. 130–137.
[181] L. Zhang, X. Xie, K. Xie, Z. Wang, Y. Lu, and Y. Zhang, "An efficient log parsing algorithm based on heuristic rules," in *Proc. Int. Symp. Adv. Parallel Process. Technol.*, 2019, pp. 123–134.
[182] L. Diederichsen, K.-K. R. Choo, and N.-A. Le-Khac, "A graph database-based approach to analyze network log files," in *Proc. Int. Conf. Netw. Syst. Secur.*, 2019, pp. 53–73.
[183] Y. Sun, S. Guo, and Z. Chen, "Intelligent log analysis system for massive and multi-source security logs: MMSLAS design and implementation plan," in *Proc. 15th Int. Conf. Mobile Ad-Hoc Sensor Netw. (MSN)*, Dec. 2019, pp. 416–421.
[184] J. Lee, Y. Lee, M. Jin, J. Kim, and J. Hong, "Analysis of application installation logs on Android systems," in *Proc. 34th ACM/SIGAPP Symp. Appl. Comput.*, Apr. 2019, pp. 2140–2145.
[185] M. Cinque, R. Della Corte, and A. Pecchia, "Microservices monitoring with event logs and black box execution tracing," *IEEE Trans. Serv. Comput.*, vol. 15, no. 1, pp. 294–307, Jan. 2019.
[186] D. Teixeira, L. Assunção, T. Pereira, S. Malta, and P. Pinto, "OSSEC IDS extension to improve log analysis and override false positive or negative detections," *J. Sensor Actuator Netw.*, vol. 8, no. 3, p. 46, Sep. 2019.
[187] C. Wang, T. Zhao, and X. Mo, "The extraction of security situation in heterogeneous log based on str-FSFDP density peak cluster," *Int. J. Comput. Sci. Eng.*, vol. 20, no. 3, p. 387, 2019.
[188] D. Walsh, P. Clough, M. M. Hall, F. Hopfgartner, J. Foster, and G. Kontonatsios, "Analysis of transaction logs from National Museums Liverpool," in *Proc. Digit. Libraries Open Knowl.*, 2019, pp. 84–98.
[189] J. You, X. Wang, L. Jin, and Y. Zhang, "Anomaly detection in the web logs using user-behaviour networks," *Int. J. Web Eng. Technol.*, vol. 14, no. 2, p. 178, 2019.
[190] C. M. Rosenberg and L. Moonen, "On the use of automated log clustering to support effort reduction in continuous engineering," in *Proc. 25th Asia–Pacific Softw. Eng. Conf. (APSEC)*, Dec. 2018, pp. 179–188.
[191] M. C. Pichiliani, "Brain controlled interface log analysis in real time strategy game matches," in *Proc. Int. Conf. Universal Access Hum.-Comput. Interact.*, 2018, pp. 256–272.
[192] Y. Nozaki and T. Satoh, "Search log analysis method of online shopping sites for navigating item categories," in *Proc. 20th Int. Conf. Inf. Integr. Web-Based Appl. Services*, Nov. 2018, pp. 87–95.
[193] B. Mansouri, M. S. Zahedi, R. Campos, and M. Farhoodi, "Online job search: Study of users' search behavior using search engine query logs," in *Proc. 41st Int. ACM SIGIR Conf. Res. Develop. Inf. Retr.*, Jun. 2018, pp. 1185–1188.
[194] M. Leemans, W. M. P. van der Aalst, and M. G. J. van den Brand, "The statechart workbench: Enabling scalable software event log analysis using process mining," in *Proc. IEEE 25th Int. Conf. Softw. Anal., Evol. Reeng. (SANER)*, Mar. 2018, pp. 502–506.
[195] M. C. Liu, C. H. Yu, J. Wu, A. C. Liu, and H. M. Chen, "Applying learning analytics to deconstruct user engagement by using log data of MOOCs," *J. Inf. Sci. Eng.*, vol. 34, no. 5, pp. 1175–1186, 2018.
[196] X. Li, Y. Wang, H. Feng, and W. Ke, "A parallel host log analysis approach based on spark," in *Proc. 14th Int. Conf. Comput. Intell. Secur. (CIS)*, Nov. 2018, pp. 301–305.
[197] H. Park, E. Kwon, S. Byon, E.-S. Jung, Y.-T. Lee, and G.-Y. Kim, "Multi-log analysis of vehicle accidents for public safety services," in *Proc. Int. Conf. Inf. Commun. Technol. Converg. (ICTC)*, Oct. 2018, pp. 1040–1042.
[198] M. Hickman, D. Fulp, E. Baseman, S. Blanchard, H. Greenberg, W. Jones, and N. DeBardeleben, "Enhancing HPC system log analysis by identifying message origin in source code," in *Proc. IEEE Int. Symp. Softw. Rel. Eng. Workshops (ISSREW)*, Oct. 2018, pp. 100–105.
[199] G. Baudart, L. Mandel, O. Tardieu, and M. Vaziri, "A reactive language for analyzing cloud logs," in *Proc. 5th ACM SIGPLAN Int. Workshop Reactive Event-Based Lang. Syst.*, Nov. 2018, pp. 61–70.
[200] S.-H. Kim, D.-H. Kim, and D.-S. Kim, "Design of smart platform-based event-log analysis and parameter-modifying software for naval combat systems," *IEIE Trans. Smart Process. Comput.*, vol. 7, no. 5, pp. 385–391, Oct. 2018.
[201] B. H. Park, Y. Hui, S. Boehm, R. A. Ashraf, C. Layton, and C. Engelmann, "A big data analytics framework for HPC log data: Three case studies using the Titan supercomputer log," in *Proc. IEEE Int. Conf. Cluster Comput. (CLUSTER)*, Sep. 2018, pp. 571–579.
[202] H. Amar, L. Bao, N. Busany, D. Lo, and S. Maoz, "Using finite-state models for log differencing," in *Proc. 26th ACM Joint Meeting Eur. Softw. Eng. Conf. Symp. Found. Softw. Eng.*, Oct. 2018, pp. 49–59.
[203] F. Ertam and M. Kaya, "Classification of firewall log files with multiclass support vector machine," in *Proc. 6th Int. Symp. Digit. Forensic Secur. (ISDFS)*, Mar. 2018, pp. 1–4.
[204] H. Zhang, W. Huang, and J. Yang, "Design and implementation of real-time log analysis system of map world platform," *Int. Arch. Photogramm., Remote Sens. Spatial Inf. Sci.*, vol. 4, pp. 745–749, Sep. 2018.

[205] J. Dongo, C. Mahmoudi, and F. Mourlin, "NDN log analysis using big data techniques: NFD performance assessment," in *Proc. IEEE 4th Int. Conf. Big Data Comput. Service Appl.*, Mar. 2018, pp. 169–175.

[206] D. D. Mishra, S. Pathan, and C. Murthy, "Apache spark based analytics of squid proxy logs," in *Proc. IEEE Int. Conf. Adv. Netw. Telecommun. Syst. (ANTS)*, Dec. 2018, pp. 1–6.

[207] V. Bhosale, A. Thakar, C. Pandit, A. Deshpande, and H. Khanuja, "Hadoop in action: Building a generic log analyzing system," in *Proc. 4th Int. Conf. Comput. Commun. Control Autom. (ICCUBEA)*, Aug. 2018, pp. 1–7.

[208] D. Castro and M. Schots, "Analysis of test log information through interactive visualizations," in *Proc. 26th Conf. Program Comprehension*, May 2018, pp. 156–166.

[209] S. Messaoudi, A. Panichella, D. Bianculli, L. Briand, and R. Sasnauskas, "A search-based approach for accurate identification of log message formats," in *Proc. 26th Conf. Program Comprehension*, May 2018, pp. 167–170.

[210] S.-H. Kim, D.-H. Kim, and D.-S. Kim, "Event log analysis software design for naval combat system using smart platform," in *Proc. Int. Conf. Electron., Inf., Commun. (ICEIC)*, Jan. 2018, pp. 1–12.

[211] S. J. Son and Y. Kwon, "Performance of ELK stack and commercial system in security log analysis," in *Proc. IEEE 13th Malaysia Int. Conf. Commun. (MICC)*, Nov. 2017, pp. 187–190.

[212] I. Y. M. Al-Mahbashi, M. B. Potdar, and P. Chauhan, "Network security enhancement through effective log analysis using ELK," in *Proc. Int. Conf. Comput. Methodol. Commun. (ICCMC)*, Jul. 2017, pp. 566–570.

[213] J. P. Gil, N. Miranda, M. Garces, and J. Avarias, "Current status of software log analysis at ALMA Observatory," *Proc. SPIE Softw. Cyber-infrastruct. Astron.*, vol. 10707, Jul. 2018, Art. no. 1070714.

[214] M. Farshchi, I. Weber, R. Della Corte, A. Pecchia, M. Cinque, J.-G. Schneider, and J. Grundy, "Contextual anomaly detection for a critical industrial system based on logs and metrics," in *Proc. 14th Eur. Dependable Comput. Conf. (EDCC)*, Sep. 2018, pp. 140–143.

[215] V. Kumar and R. S. Thakur, *Web Log Analysis Tools: At a Glance*, vol. 34. Singapore: Springer, 2018.

[216] N. N. Y. Vo, S. Liu, J. Brownlow, C. Chu, B. Culbert, and G. Xu, "Client churn prediction with call log analysis," in *Proc. Database Syst. Adv. Appl.*, 2018, pp. 752–763.

[217] K. M. Matthew and A. Quadir Md, "Analysis framework for logs in communication devices," *Int. J. Web Portals*, vol. 10, no. 1, pp. 15–26, Jan. 2018.

[218] K. Kurniawan, "Semantic query federation for scalable security log analysis," in *Proc. Eur. Semantic Web Conf.*, Jun. 2018, vol. 11155, pp. 294–303.

[219] Y. Su, J. Li, D. Song, P. Zhang, and Y. Zhang, "Investigating the dynamic decision mechanisms of users' relevance judgment for information retrieval via log analysis," in *Proc. Pacific Rim Int. Conf. Artif. Intell.*, 2018, pp. 968–979.

[220] H. Park, E. Kwon, E.-S. Jung, S. Byon, H.-W. Lee, and Y.-T. Lee, "Multi-log analysis platform for supporting public safety service," in *Proc. Int. Conf. Inf. Commun. Technol. Converg. (ICTC)*, 2017, pp. 1137–1139.

[221] K. Otomo, S. Kobayashi, K. Fukuda, and H. Esaki, "An analysis of burstiness and causality of system logs," in *Proc. Asian Internet Eng. Conf.*, Nov. 2017, pp. 16–23.

[222] J. El Abdelkhalki, M. Ben Ahmed, and B. H. Anouar, "Classification and exploration of TSM log file based on datamining algorithms," in *Proc. 2nd Int. Conf. Comput. Wireless Commun. Syst.*, 2017, pp. 1–7.

[223] E. E.-D. Hemdan and D. H. Manjaiah, "Spark-based log data analysis for reconstruction of cybercrime events in cloud environment," in *Proc. Int. Conf. Circuit, Power Comput. Technol. (ICCPCT)*, Apr. 2017, pp. 1–8.

[224] P. He, J. Zhu, Z. Zheng, and M. R. Lyu, "Drain: An online log parsing approach with fixed depth tree," in *Proc. IEEE 24th Int. Conf. Web Serv.*, 2017, pp. 33–40.

[225] C. Vega, P. Roquero, R. Leira, I. Gonzalez, and J. Aracil, "Loginson: A transform and load system for very large-scale log analysis in large IT infrastructures," *J. Supercomput.*, vol. 73, no. 9, pp. 3879–3900, Sep. 2017.

[226] Y. Li, B. Chen, V. W. Zheng, W. G. Temple, Z. Kalbarczyk, and Y. Wu, "Enhancing anomaly diagnosis of automatic train supervision system based on operation log," in *Proc. 47th Annu. IEEE/IFIP Int. Conf. Dependable Syst. Netw. Workshops (DSN-W)*, Jun. 2017, pp. 133–136.

[227] T. Li, Y. Jiang, C. Zeng, B. Xia, Z. Liu, W. Zhou, X. Zhu, W. Wang, L. Zhang, J. Wu, L. Xue, and D. Bao, "FLAP: An End-to-End event log analysis platform for system management," in *Proc. 23rd ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Aug. 2017, pp. 1547–1556.

[228] F. C. Liu, W. Xu, M. Belgin, R. Huang, and B. C. Fleischer, "Insights into research computing operations using big data-powered log analysis," in *Proc. Pract. Exper. Adv. Res. Comput. Sustainability, Success Impact*, Jul. 2017, pp. 1–8.

[229] P. W. D. C. Jayathilake, N. R. Weeraddana, and H. K. E. P. Hettiarachchi, "Automatic detection of multi-line templates in software log files," in *Proc. 17th Int. Conf. Adv. ICT Emerg. Regions (ICTer)*, Sep. 2017, pp. 1–8.

[230] F. Shoeleh, M. S. Zahedi, and M. Farhoodi, "Search engine pictures: Empirical analysis of a web search engine query log," in *Proc. 3rd Int. Conf. Web Res. (ICWR)*, Apr. 2017, pp. 90–95.

[231] M. S. Zahedi, B. Mansouri, S. Moradkhani, M. Farhoodi, and F. Oroumchian, "How questions are posed to a search engine? An empiricial analysis of question queries in a large scale Persian search engine log," in *Proc. 3rd Int. Conf. Web Res. (ICWR)*, Apr. 2017, pp. 84–89.

[232] J. Raja and M. Ramakrishnan, "Implementing continuous auditing and compression technique in log auditing," in *Proc. 8th Int. Conf. Adv. Comput. (ICoAC)*, Jan. 2017, pp. 196–200.

[233] M. Dayarathna, P. Akmeemana, S. Perera, and M. Jayasinghe, "Solution recommender for system failure recovery via log event pattern matching on a knowledge graph," in *Proc. 11th ACM Int. Conf. Distrib. Event-Based Syst.*, Jun. 2017, pp. 331–334.

[234] B. D. Bryant and H. Saiedian, "A novel kill-chain framework for remote security log analysis with SIEM software," *Comput. Secur.*, vol. 67, no. Mar. 2017, pp. 198–210, 2017.

[235] D. Hienert, "User interests in German social science literature search: A large scale log analysis," in *Proc. Conf. Conf. Hum. Inf. Interact. Retr.*, Mar. 2017, pp. 1–6.

[236] I. Mavridis and H. Karatza, "Performance evaluation of cloud-based log file analysis with apache Hadoop and apache spark," *J. Syst. Softw.*, vol. 125, pp. 133–151, Mar. 2017.

[237] R. Rastogi, S. Akash, G. Shobha, G. Poonam, D. Pratiba, and A. Singh, "Design and development of generic web based framework for log analysis," in *Proc. IEEE Region 10 Conf.*, Nov. 2016, pp. 232–236.

[238] J. Tong, L. Ying, T. Hongyan, and W. Zhonghai, "An approach to pinpointing bug-induced failure in logs of open cloud platforms," in *Proc. IEEE 9th Int. Conf. Cloud Comput. (CLOUD)*, Jun. 2016, pp. 294–302.

[239] J. I. Jang, K. H. Kim, N. Kim, G. Cho, and M. Kim, "A log handling application framework for process mining of steel industry," *ICIC Exp. Lett., B, Appl.*, vol. 8, no. 1, pp. 43–50, 2017.

[240] E. Kacprzak, L. M. Koesten, E. Simperl, and J. Tennison, "A query log analysis of dataset search," in *Proc. Int. Conf. Web Eng.*, 2017, pp. 429–436.

[241] Q. Ai, S. T. Dumais, N. Craswell, and D. Liebling, "Characterizing email search using large-scale behavioral logs and surveys," in *Proc. 26th Int. Conf. World Wide Web*, Apr. 2017, pp. 1511–1520.

[242] Y. Kanno, K. Kanamori, and H. Nishiyama, "Access log analysis for increasing the number of applicants using LCS on a recruitment site," in *Proc. 32nd Int. Conf. Comput. Appl.*, 2017, p. 23.

[243] Z. Wu, X. Xie, Y. Liu, M. Zhang, and S. Ma, "A study of user image search behavior based on log analysis," in *Proc. China Conf. Inf. Retr.*, 2017, pp. 69–80.

[244] J. J. Kim and J. Park, "R based network log analysis techniques," *Information*, vol. 20, no. 10, pp. 7843–7856, 2017.

[245] J. Wang, C. Li, S. Han, S. Sarkar, and X. Zhou, "Predictive maintenance based on event-log analysis: A case study," *IBM J. Res. Develop.*, vol. 61, no. 1, pp. 121–132, Jan. 2017.

[246] M. M. Saudi, F. Ridzuan, and H. A. B. Hashim, "An efficient data transformation technique for web log," *Lect. Notes Eng. Comput. Sci.*, vol. 2229, pp. 434–439, Oct. 2017.

[247] A. Awad, N. M. Zaki, and C. Di Francescomarino, "Analyzing and repairing overlapping work items in process logs," *Inf. Softw. Technol.*, vol. 80, pp. 110–123, Nov. 2016.

[248] A. Balliu, D. Olivetti, O. Babaoglu, M. Marzolla, and A. Sirbu, "A big data analyzer for large trace logs," *Computing*, vol. 98, no. 12, pp. 1225–1249, Dec. 2016.

[249] T.-T. Wu, "A learning log analysis of an english-reading e-book system combined with a guidance mechanism," *Interact. Learn. Environ.*, vol. 24, no. 8, pp. 1938–1956, Nov. 2016.

[250] M. A. Mercado-Varela, A. García-Holgado, F. J. García-Peñalvo, and M. S. Ramírez-Montoya, "Analyzing navigation logs in MOOC: A case study," in *Proc. 4th Int. Conf. Technol. Ecosyst. Enhancing Multiculturality*, Nov. 2016, pp. 873–880.
[251] W. Peng, Y. Li, B. Li, and X. Zhu, "An analysis platform of road traffic management system log data based on distributed storage and parallel computing techniques," in *Proc. IEEE Int. Conferences Big Data Cloud Comput.*, Oct. 2016, pp. 585–589.
[252] O. ElTayeby and W. Dou, "A survey on interaction log analysis for evaluating exploratory visualizations," in *Proc. Beyond Time Errors Novel Eval. Methods Vis.*, 2016, pp. 62–69.
[253] D. A. Girei, M. Ali Shah, and M. B. Shahid, "An enhanced botnet detection technique for mobile devices using log analysis," in *Proc. 22nd Int. Conf. Autom. Comput. (ICAC)*, Sep. 2016, pp. 450–455.
[254] H. Chen, S. Tu, C. Zhao, and Y. Huang, "Provenance cloud security auditing system based on log analysis," in *Proc. Int. Conf. Online Anal. Comput. Sci. (ICOACS)*, 2016, pp. 155–159.
[255] G. Meera and G. Geethakumari, "Event correlation for log analysis in the cloud," in *Proc. IEEE 6th Int. Conf. Adv. Comput. (IACC)*, Feb. 2016, pp. 158–162.
[256] M. Wurzenberger, F. Skopik, G. Settanni, and W. Scherrer, "Complex log file synthesis for rapid sandbox-benchmarking of security- and computer network analysis tools," *Inf. Syst.*, vol. 60, pp. 13–33, Dec. 2016.
[257] R. Huang, W. Xu, and R. McLay, "A web interface for XALT log data analysis," in *Proc. Conf. Diversity, Big Data, Sci. Scale*, Jul. 2016, pp. 1–8.
[258] T. Prakash, M. Kakkar, and K. Patel, "Geo-identification of web users through logs using ELK stack," in *Proc. 6th Int. Conf. Cloud Syst. Big Data Eng.*, Jan. 2016, pp. 606–610.
[259] N. Busany and S. Maoz, "Behavioral log analysis with statistical guarantees," in *Proc. 38th Int. Conf. Softw. Eng.*, May 2016, pp. 877–887.
[260] D. Nacu, C. K. Martin, M. Schutzenhofer, and N. Pinkard, "Beyond traditional metrics: Using automated log coding to understand 21st century learning online," in *Proc. 3rd ACM Conf. Learn. Scale*, Apr. 2016, pp. 197–200.
[261] J. Palotti, A. Hanbury, H. Müller, and C. E. Kahn, "How users search and what they search for in the medical domain: Understanding laypeople and experts through query logs," *Inf. Retr.*, vol. 19, nos. 1–2, pp. 189–224, 2016.
[262] M. Hinkka, T. Lehto, and K. Heljanko, "Assessing big data SQL frameworks for analyzing event logs," in *Proc. 24th Euromicro Int. Conf. Parallel, Distrib., Network-Based Process. (PDP)*, Feb. 2016, pp. 101–108.
[263] A. R. Hussain, M. A. Hameed, and S. Fatima, "A proposal: High-throughput robust architecture for log analysis and data stream mining," in *Adv. Intell. Syst. Comput. Innov. Comput. Sci. Eng.*, 2016, pp. 305–314.
[264] K. B. Naukudkar, D. D. Ambawade, and J. W. Bakal, "Enhancing performance of security log analysis using correlation-prediction technique," in *Proc. Int. Conf. ICT Sustain. Develop.*, 2016, pp. 635–643.
[265] J. P. Gil, J. Reveco, and T.-C. Shen, "Operational logs analysis at ALMA observatory based on ELK stack," *Proc. Softw. Cyberinfrastruct. Astron.*, vol. 9913, pp. 814–822, Jun. 2016.
[266] X. Zhou, P. Zhang, and J. Wang, "Examining task relationships in multitasking consumer search sessions: A query log analysis," *Assoc. Inf. Sci. Technol.*, vol. 53, no. 1, pp. 1–5, 2016.
[267] P. Braslavski, V. Petras, and V. Likhosherstov, "Ten months of digital reading: An exploratory log study," in *Proc. Int. Conf. Theory Pract. Digit. Libraries*, 2016, pp. 392–397.
[268] J. Wajda and W. Zadrozny, "Prior-art relevance ranking based on the examiner's query log content," in *Challenging Problems and Solutions in Intelligent Systems Studies in Computational Intelligence*. Cham, Switzerland: Springer, 2016, pp. 323–333.
[269] A. M. M. Morais, L. G. D. Vasconcelos, and R. D. C. Santos, "Challenges in mapping behaviours to activities using logs from a citizen science project," *Proc. SPIE Next-Gener. Anal.*, vol. 9851, pp. 118–132, May 2016.
[270] D. Suryanarayana, P. Kanakam, S. M. Hussain, and S. Gupta, "Cognitive analytic task based on search query logs for semantic identification," *Int. J. Control Theory Appl.*, vol. 9, no. 21, pp. 273–280, 2016.
[271] H. Au, M. Diallo, and K. Lee, "Multi-stage analysis of intrusion detection logs for quick impact assessment," in *Proc. Eur. Conf. Inf. Warfare Secur.*, 2016, p. 18.
[272] *Log Analysis from A to Z: A Literature Survey*. Accessed: Sep. 5, 2021. [Online]. Available: https://repository.tudelft.nl/islandora/object/uuid:90afad8d-7010-4407-a502-fb5d73c0f291/datastream/OBJ/download
[273] D. J. Chalyy, N. I. Ovchenkov, E. G. Lazareva, and R. R. Yaikov, "A security system event log analysis," in *Proc. CEUR Workshop*, 2018, vol. 2268, pp. 141–146.
[274] D. Nicholas and P. Huntington, "Micro-mining and segmented log file analysis: A method for enriching the data yield from internet log files," *J. Inf. Sci.*, vol. 29, no. 5, pp. 391–404, Oct. 2003.
[275] B. J. Jansen and A. Spink, "How are we searching the world wide web? A comparison of nine search engine transaction logs," *Inf. Process. Manag.*, vol. 42, no. 1, pp. 248–263, 2006.
[276] J. Teevan, E. Adar, R. Jones, and M. Potts, "History repeats itself: Repeat queries in Yahoo's logs," in *Proc. 29th Annu. Int. ACM SIGIR Conf. Res. Develop. Inf. Retr.*, 2006, pp. 703–704.
[277] D. Nicholas, P. Huntington, H. R. Jamali, and C. Tenopir, "What deep log analysis tells us about the impact of big deals: Case study OhioLINK," *J. Documentation*, vol. 62, no. 4, pp. 482–508, Jul. 2006.
[278] J. Teevan, E. Adar, R. Jones, and M. A. S. Potts, "Information re-retrieval: Repeat queries in Yahoo's logs," in *Proc. 30th Annu. Int. ACM SIGIR Conf. Res. Develop. Inf. Retr.*, 2007, pp. 151–158.
[279] H. Lam, D. Russell, D. Tang, and T. Munzner, "Session viewer: Visual exploratory analysis of web session logs," in *Proc. IEEE Symp. Vis. Anal. Sci. Technol.*, Oct. 2007, pp. 147–154.
[280] R. Vaarandi, "Mining event logs with SLCT and LogHound," in *Proc. NOMS - IEEE Netw. Oper. sManage. Symp.*, 2008, pp. 1071–1074.
[281] D. Nicholas, P. Huntington, and H. R. Jamali, "User diversity: As demonstrated by deep log analysis," *Electron. Library*, vol. 26, no. 1, pp. 21–38, Feb. 2008.
[282] D. J. Brenes and D. Gayo-Avello, "Stratified analysis of AOL query log," *Inf. Sci.*, vol. 179, no. 12, pp. 1844–1858, May 2009.
[283] S. K. Tyler and J. Teevan, "Large scale query log analysis of re-finding," in *Proc. 3rd ACM Int. Conf. Web Search Data Mining*, 2010, pp. 191–200.
[284] R. W. White and J. Huang, "Assessing the scenic route: Measuring the value of search trails in web logs," in *Proc. 33rd Int. ACM SIGIR Conf. Res. Develop. Inf. Retr.*, 2010, pp. 587–594.
[285] S. Duarte Torres, D. Hiemstra, and P. Serdyukov, "Query log analysis in the context of information retrieval for children," in *Proc. 33rd Int. ACM SIGIR Conf. Res. Develop. Inf. Retr.*, 2010, pp. 847–848.
[286] Z. Zheng, L. Yu, W. Tang, Z. Lan, R. Gupta, N. Desai, S. Coghlan, and D. Buettner, "Co-analysis of RAS log and job log on Blue Gene/P," in *Proc. IEEE Int. Parallel Distrib. Process. Symp.*, May 2011, pp. 840–851.
[287] D. Elsweiler, M. Harvey, and M. Hacker, "Understanding re-finding behavior in naturalistic email interaction logs," in *Proc. 34th Int. ACM SIGIR Conf. Res. Develop. Inf.*, 2011, pp. 35–44.
[288] W. Weerkamp, R. Berendsen, B. Kovachev, E. Meij, K. Balog, and M. de Rijke, "People searching for people: Analysis of a people search engine log," in *Proc. 34th Int. ACM SIGIR Conf. Res. Develop. Inf.*, 2011, pp. 45–54.
[289] A. Makanju, A. N. Zincir-Heywood, and E. E. Milios, "A lightweight algorithm for message type extraction in system application logs," *IEEE Trans. Knowl. Data Eng.*, vol. 24, no. 11, pp. 1921–1936, Nov. 2012.
[290] S. K. Bajracharya and C. V. Lopes, "Analyzing and mining a code search engine usage log," *Empir. Softw. Eng.*, vol. 17, nos. 4–5, pp. 424–466, 2012.
[291] C. Di Martino, M. Cinque, and D. Cotroneo, "Assessing time coalescence techniques for the analysis of supercomputer logs," in *Proc. IEEE/IFIP Int. Conf. Dependable Syst. Netw.*, Jun. 2012, pp. 1–12.
[292] J. Lu and N. W. Y. Law, "Understanding collaborative learning behavior from moodle log data," *Interact. Learn. Environ.*, vol. 20, no. 5, pp. 451–466, Oct. 2012.
[293] X. Lin, P. Wang, and B. Wu, "Log analysis in cloud computing environment with Hadoop and spark," in *Proc. 5th IEEE Int. Conf. Broadband Netw. Multimedia Technol.*, Nov. 2013, pp. 273–276.
[294] M. P. Kato, T. Sakai, and K. Tanaka, "When do people use query suggestion? A query suggestion log analysis," *Inf. Retr.*, vol. 16, no. 6, pp. 725–746, Dec. 2013.
[295] R. Vaarandi and M. Pihelgas, "LogCluster—A data clustering and pattern mining algorithm for event logs," in *11th Int. Conf. Netw. Service Manage. (CNSM)*, 2015, pp. 1–7.
[296] A. Juvonen and T. Sipola, "Online anomaly detection using dimensionality reduction techniques for HTTP log analysis," *Comput. Netw.*, vol. 91, pp. 46–56, Oct. 2015.
[297] S. Bhatia, D. Majumdar, and P. Mitra, "Query suggestions in the absence of query logs," in *Proc. 34th Int. ACM SIGIR Conf. Res. Develop. Inf.*, 2011, pp. 795–804.

[298] C. Lucchese, S. Orlando, R. Perego, F. Silvestri, and G. Tolomei, "Identifying task-based sessions in search engine query logs," in *Proc. 4th ACM Int. Conf. Web Search Data Mining*, 2011, pp. 277–286.
[299] C. Lucchese, S. Orlando, R. Perego, F. Silvestri, and G. Tolomei, "Discovering tasks from search engine query logs," *ACM Trans. Inf. Syst.*, vol. 31, no. 3, pp. 1–43, 2013.

ŁUKASZ KORZENIOWSKI received the M.Sc. degree in computer science from the Gdańsk University of Technology, Gdańsk, Poland, in 2005, where he is currently pursuing the Ph.D. degree in computer science.

Since 2005, he has been working as a software engineer in multiple industries, including telecommunications, healthcare, and workforce management. Since 2014, he has been working at Nordea Bank Abp developing solutions for the finance sector. His research interests include automated means of extracting knowledge about software systems, digital transformation of legacy systems, systems' explainability, and automated programming.

KRZYSZTOF GOCZYŁA graduated from the Faculty of Electronics, Gdańsk University of Technology (GUT), in 1976. He received the Ph.D. and Habilitation degrees in computer science, in 1982 and 1999, respectively.

In 2012, he received the title of a Professor in technical sciences. He is currently the Head of the Department of Software Engineering, GUT. Until now, he has promoted seven doctors of technical sciences. He co-founded at GUT the "data engineering" field of study and the "engineering information systems" M.Sc. study option. He conducts courses related to software engineering, databases, data warehouses, and knowledge bases. The field of knowledge management and Semantic Web technologies has been the subject of several projects implemented under his direction. He is the author of the original indexing method for complex data in non-relational databases called partial-order trees. He has authored or coauthored over 170 scientific publications, including four books. The areas of his current research interests include databases and knowledge bases, data warehouses, software engineering, and knowledge engineering. His research interests include non-relational database models, big data management, object-oriented software engineering, and management of data and knowledge.