Array 13 (2022) 100120

Contents lists available at ScienceDirect

Array

journal homepage: www.elsevier.com/locate/array

# Process mining usage in cybersecurity and software reliability analysis: A systematic literature review

Martin Macak [a,*], Lukas Daubner [a], Mohammadreza Fani Sani [b], Barbora Buhnova [a]

[a] Faculty of Informatics, Masaryk University, Botanická 68a, 602 00 Brno, Czechia
[b] Process and Data Science Chair, RWTH-Aachen University, Aachen, Germany

ARTICLE INFO

*Keywords:*
Process mining
Cybersecurity
Software reliability
Systematic literature review

ABSTRACT

The digitalization of our society is only possible in the presence of secure and reliable software systems governing ongoing critical processes, so-called critical information infrastructures. The understanding of mutual interdependencies of events and processes is crucial for cybersecurity and software reliability. One of the promising ways to tackle these challenges is process mining, which is a set of techniques that aims to mine essential knowledge from processes, thus providing more perspectives and temporal context to data interpretation and process understanding. However, it is unclear how process mining can help and can be practically used in the context of cybersecurity and reliability.

Therefore, in this work, we investigate the potential of process mining to aid in cybersecurity and software reliability to analyze and support research efforts in these areas. Concretely, we collect existing process mining applications, discuss current trends and promising research directions that can be used to tackle the current cybersecurity and software reliability challenges. To this end, we conduct a systematic literature review covering 35 relevant research approaches to examine how the process mining is currently used for these tasks and what are the research gaps and promising research directions in the area. This work is an extension of our previous work, which focused solely on the cybersecurity area, based on the observation of relative closeness and similar goals of those two fields, in which some approaches tend to overlap.

## 1. Introduction

The advancement of digitalization in modern society has fueled the role of cybersecurity and software reliability in various domains of critical information infrastructures, such as healthcare or transportation, where potential problems could result in injuries or loss of lives. Nowadays, the key challenge of effective cybersecurity and software reliability assurance is the actual rapid advancement of information technology, with new types of threats and unprecedented discrepancies emerging daily.

While both cybersecurity and software reliability are concerned with different root causes, ultimately, they overlap in their end goal of assuring the availability and integrity of cyber systems [1]. For example, a system can be rendered unavailable by both a successful cyberattack or software bug. Likewise, a system or its data can be inappropriately modified by a malicious insider or accidental misconfiguration. As such, a holistic approach has to consider the aspects of both fields [2].

Existing cybersecurity and reliability techniques are designed for the discovery and prevention of a specific type of problem, and hence having difficulties in adapting to new threats [3,4]. Furthermore, the actual security and reliability threats develop over time within complex processes in which minor vulnerabilities (e.g., software bugs, weak separation of authenticated spaces) combine with human/operator errors (e.g., credentials leaks) into major problems that are challenging to detect in its early formation stages [5]. Hence, the investigation of security threats is still largely manual [6] or is being addressed with strongly specialized domain-specific techniques to reduce false positives [7,8]. The behavior of entities is often encoded into a mathematical model that might be hard to manipulate, abstract, or complex, making it hard to respond to the security threats adequately [9].

Process mining [10] is a set of techniques that could be promising in addressing the aforementioned challenges. In contrast to traditional data-centric approaches (like data mining) and process-centric approaches (such as BPM analysis), process mining involves both data and end-to-end processes in their analysis [11,12] to the benefit of the final results [13,14]. For example, process mining techniques are designed to examine when and how a process deviated from the designed

* Corresponding author.
*E-mail addresses:* macak@mail.muni.cz (M. Macak), daubner@mail.muni.cz (L. Daubner), fanisani@pads.rwth-aachen.de (M. Fani Sani), buhnova@mail.muni.cz (B. Buhnova).

https://doi.org/10.1016/j.array.2021.100120
Received 26 August 2021; Received in revised form 18 November 2021; Accepted 4 December 2021
Available online 22 December 2021

process model or how a bottleneck activity results in the final delays of a service/product. Moreover, it is designed to discover, monitor, and enhance processes by extracting knowledge from event data (i.e., *event logs*). Process mining has already proven successful in many domains, aiding with challenging tasks such as fraud detection [15], robotic process automation [16], or learning analytics [17]. Furthermore, process mining became also favored in governing safety-critical processes, such as in healthcare, where it supports critical hospital processes and patient treatment [18]. In [19], it is explained why process mining can be beneficial for advanced analysis and provides several use cases of it in different fields.

The evidence-based benefits and capabilities of process mining in similar domains such as software engineering [20] and confidentiality [21] make it a promising candidate to address the challenges in cybersecurity and software reliability analysis — using its techniques, we might be able to more effectively detect unexpected behavior [22], identify issues [23], detect deviations [24], or verify whether the system conforms to the designed process [25]. Furthermore, it might have the capability to offer an overview of alerts in the system, detect malware in a system, detect frauds, verify the user behavior, or identify software bugs. This brings new opportunities of process mining to advance the state of research in the cybersecurity and software-reliability situations with uncertainty about the actual processes underlying runtime system behavior, which needs to be reconstructed based on the observed events in the system.

Currently, there is no comprehensive systematic literature review that would help the researchers and practitioners understand where and how exactly the process mining could aid in these specific situations.

This paper aims to enhance researchers' efforts in cybersecurity and software reliability areas via an overview of research approaches using process mining for the purpose of cybersecurity and software reliability in various domains and for various tasks. The papers are grouped by their application domain to give insight into the current progress of process-mining usage in each one of them. We identify the techniques that are used for this purpose, together with their properties, and discuss possible research directions for further progress in this area. It is an enhancement to our previous study [26], which was limited in scope to the cybersecurity area. The rationale behind the inclusion of the software reliability into the systematic literature review is the observation of relative closeness, overlap in techniques, and similar goals of those two fields when conducting the original review. For this reason, we aim to explore the broader scope, as the techniques might support each other based on the overlap. Furthermore, the combined study should provide greater insight into the field for researchers as well.

The remainder of this paper is structured as follows. Section 2 contains a background of process mining. Section 3 provides an overview of existing literature reviews on process mining in other areas. Then, in Section 4, we formulate the research questions and describe our methodology. The usage of process mining to ensure cybersecurity is then detailed in Section 5, followed by Section 6, which focuses on the application of process mining for software reliability. The results of the review and the answers to our research questions are in Section 7. Section 8 discusses the threats to the validity of our review. Finally, Section 9 concludes the paper.

## 2. Background

Process mining techniques have proved to be very successful in (1) *process discovery*, which aims to find a descriptive model of the underlying process from event logs, (2) *conformance checking*, i.e., monitoring and inspecting whether the real execution of the process conforms to the corresponding designed (or discovered) reference process model, and (3) *process enhancement*, which improves and enriches a process model based on the related event data [10].

Process discovery is able to find a model that represents the process described in the event log. This model has to conform to four quality criteria — fitness, precision, generalization, and simplicity [10]. The model has a low fitness when it can replay only a small number of traces in the event log. When the model has poor precision, it means that it allows a very different behavior from the behavior in the log. On the other hand, the model with a low generalization allows only the behavior that was in the log. Simplicity of the model is connected to whether the model explains the behavior with the minimum necessary information. Process discovery has been first discussed in [27], which describes discovery methods in the context of software engineering processes. Similar to some later published techniques [28,29], it was limited to sequential processes. One of the first discovery algorithms that handled concurrency of events is the Alpha algorithm [30]. It produces a marked Petri net from an event log. Later, many other algorithms emerged, like variants of Alpha algorithm [31], Heuristic Miner [32], Fuzzy miner [33], and DecMiner [34].

The purpose of conformance checking is to decide whether the execution of the process conforms to the corresponding process model [35]. Early conformance checking techniques used token-based replay to detect non-fitting cases. They replayed a trace of events in a Petri net, and based on it, produced a diagnostics [10]. For example, Conformance Checker [36] introduced two metrics: fitness and appropriateness. Fitness measures the degree to which the process model can replay the traces from the log. Appropriateness measures the simplicity, precision, and generalization of the model. However, the token-based approach often does not provide satisfactory results, so other alternatives, like alignment-based solutions were introduced [37].

Process enhancement techniques aim to improve or extend an existing process model using information extracted from the process described in an event log [10]. This is important when the model does not reflect the reality accurately. The example of process improvement is [38], where the authors repair the given model, increasing its fitness with respect to the given event log. In process extension, a new perspective is added to the process model, such as an organizational or time perspective. The approach in [39] uses the organizational perspective to enhance the model by roles of the activity originators. On the other hand, in [40], the time perspective is used.

## 3. Related work

Several publications have performed a literature review on the usage of process mining. These reviews are typically domain-specific, with healthcare [41,42] and education [43] being the most popular domains. In [18], Garcia et al. employed a general multi-domain systematic literature review to map the applications of process mining. They identified 19 application domains of process mining and sorted them by the number of published papers. For the top six areas, i.e., healthcare, ICT, manufacturing, education, finance, and logistics, they described the main contribution of process mining. As the security was placed in eighth place among the domains, the detail of its contributions was not discussed. Moreover, reliability has not been considered at all.

Within the healthcare domain, Kurniati et al. [41] performed a literature review of process mining usage in oncology. Moreover, Williams et al. [42] reviewed the application of process mining in primary care, and Yang and Su [44] reviewed the studies on process mining techniques for clinical pathways. Lastly, Rojas et al. [45] performed a more general literature review about the applications of process mining in the healthcare domain.

Within the education domain, Bogarin et al. [43] did a review of the publications that utilize process mining for the analysis of educational processes.

Within the security domain, there is no comprehensive review of process mining available yet. Some pieces of the work in this area can

**Table 1**
Summary of review protocol.

| | | |
|---|---|---|
| Initial security papers | IEEE Xplore | 33 |
| | ScienceDirect | 272 |
| | Springer Link | 350 |
| | ACM DL | 97 |
| | Web of Science | 29 |
| | Scopus | 79 |
| | **Total** | **860** |
| Initial reliability papers | IEEE Xplore | 28 |
| | ScienceDirect | 835 |
| | Springer Link | 524 |
| | ACM DL | 274 |
| | Web of Science | 62 |
| | Scopus | 110 |
| | **Total** | **1833** |
| Unique papers | | 1967 |
| Filtered papers | 1st Stage — title & abstract | 121 |
| | 2nd Stage — Full text | 30 |
| Snowballed papers | Google search | 2 |
| | References | 3 |
| Relevant papers | Security | 25 |
| | Reliability | 10 |
| | **Total** | **35** |

be found in the review by Leitner and Rinderle-Ma [46], which focuses on security in Process-Aware Information Systems (PAIS), where process mining is however only one of many methods considered. Moreover, as the review covered results from 1993 to 2012, many recent approaches are missing. In [47], Kelemen provided an overview of the usage of process mining for security in the public sector domain, published between 2000 and 2016. Based on this review, the author provides a set of topics that are dominant in the identified research papers, together with challenges and future research directions. The paper, however, only considers the public sector domain (explicitly in its inclusion criteria). Hence, our work, which is an enhancement to our previous study [26], which was limited in scope to the cybersecurity area, continues this research path with a broader study, including more recent publications and a wider spectrum of domains.

Moving away from surveys focusing on process mining, there are many surveys on specific aspects of cybersecurity or software reliability. However, they tend to focus on particular domains or techniques [48–51]. Also, an overview of existing surveys in cybersecurity [52] does not include focus on process mining.

With the missing comprehensive systematic literature review of process mining usage for cybersecurity and software reliability, it is hard to understand where and how the process mining could help the researchers and practitioners. The focus on process mining applications for these tasks in different research directions is an important aspect to consider. Moreover, none of the existing reviews covers the application of process mining in software reliability, which makes the systematic literature review presented in this paper a valuable complement of the current state of the art.

## 4. Methodology

The goal of this study is to review recent research in cybersecurity and software reliability that employs process mining. To this end, we formulate a strategy to guide this study, based on the Kitchenham and Charters guidelines for systematic literature reviews [53]. This includes a digital library search for current literature, snowballing, and manual searching. Within the review, we first consider the cybersecurity and software reliability separately and merge them in later stages. Our review methodology is visualized in Fig. 1, together with the results of each conducted search and processing stage. See Table 1 for a summary of the review protocol containing the number of publications.

**Research questions** As the first step, the following research questions had been formulated. They guide the search, inclusion and exclusion criteria, and the assessment of results.

*RQ1: What are the research directions in which the process mining is used to ensure cybersecurity and software reliability?* The aspects of cybersecurity and software reliability are usually considered within broader application domains such as networking, banking, or critical infrastructures. The goal is to identify the research directions with utilization or active research in process mining to ensure cybersecurity and software reliability.

*RQ2: Which process mining techniques and approaches are utilized to ensure cybersecurity and software reliability?* Process mining itself is a collection of different techniques, which can be used in numerous ways. For example, there is a major difference between analyzing past and present events. The goal is to identify those commonly utilized for cybersecurity and software reliability, those that are not, and those rarely considered. Comparing results from the first two research questions with the typical process mining research trends can provide valuable insights. Concretely, the focus is placed on the used technique, target period, usage of expert knowledge, and automation of the analysis.

*RQ3: What are the current gaps and possible research directions in the usage of process mining for cybersecurity and software reliability?* The last goal is combining the outcomes of the first two research questions to assess opportunities for researchers and available methods for practitioners. Mainly in comparison to the general utilization of process mining.

**Primary collection** To establish a base collection of papers, we searched in several digital libraries, specifically IEEE Xplore,[1] Elsevier ScienceDirect,[2] Springer Link,[3] ACM Digital Library,[4] Web of Science,[5] and Scopus.[6] The search was limited to the recent research over the last six years, from 2014 to 2020. Furthermore, we only considered papers written in English.

The search was divided into two parts. One for the cybersecurity and one for software reliability. The reason for this decision is that while both domains share many similar goals, their communities might differ. Regardless of the division, the results of the searches are combined in the subsequent stages of our review method, so that we take advantage of the differences as well as synergies. The search string used to search the digital libraries is in Listing 1 for cybersecurity and Listing 2 for software reliability.

Listing 1: Search string for cybersecurity

```
("process mining") AND
    (cybersecurity OR security)
```

Listing 2: Search string for software reliability

```
("process mining") AND
    (dependability OR reliability OR
     availability OR robustness OR
     resilience)
```

Thereafter, all the collected papers are further grouped and deduplicated. Results from both searches are combined because we found that around 27% of papers were covered by both cybersecurity and software reliability search strings, or multiple primary sources.

**Filtering** The gathered primary collection of papers contains a large number of papers that are not relevant to the scope of this literature

[1] https://ieeexplore.ieee.org/.
[2] https://www.sciencedirect.com/.
[3] https://link.springer.com/.
[4] https://dl.acm.org/.
[5] https://webofknowledge.com/.
[6] https://www.scopus.com/.

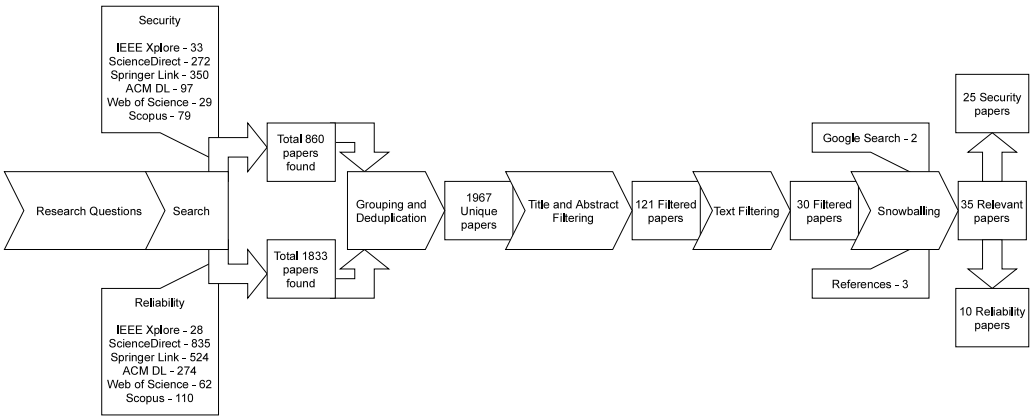


**Fig. 1.** The strategy of the review with the number of publications at each stage.

review. Therefore, the next stage was to filter out false-positive results from the collection.

There are two steps for filtering the collection. In the first step, the only parameters considered were paper title, abstract, keywords, and conclusion of the paper. The papers were labeled by their topic, which is cybersecurity or software reliability. The following second step of reading the full paper to confirm or correct the labeling. Any relabeling was confirmed with an independent opinion by another team member.

To maintain the fairness of the filtering, several inclusion and exclusion criteria were formulated. Conforming to one or more of the exclusion criteria results in discarding the paper. The paper is kept if and only if it conforms to at least one inclusion criterion. Otherwise, if it conforms to neither, it is discarded. Additionally, if there are multiple papers from the same authors that are directly related (i.e., follow-up of the same idea, but not a new topic), only the most recent or most extensive is kept. The *inclusion criteria* are that the work:

- deals with the detection, analysis, modeling, recovery, or avoidance of cyberattack, malware, fraud, fault, or error;
- focuses on cybersecurity-related processes;
- aims at the reliability of software systems;
- is concerned with anomalous behavior regarding cybersecurity or software reliability.

The *exclusion criteria* are that the work:

- does not utilize process mining;
- does not consider software systems as the primary target of the approach;
- is focused only on reliable performance/effectiveness;
- is not a full paper (i.e., is only a keynote abstract, chapter fragment, or encyclopedia article).

Each filtering step was done by a different researcher. In case of any doubts, the paper was marked and discussed by two more researchers to reach the consensus. To further avoid bias in filtering, a sample of 10% papers filtered out in the first phase was revived by another researcher.

**Snowballing** The last stage is the snowballing, which includes papers referred by papers kept after the second step of filtering. Each snowballed paper was considered in terms of inclusion and exclusion criteria, and within the 2014-to-2020 limit, and English-written condition.

The final collection of papers was further supplemented by manual, non-systematic search using search engines like Google Scholar.[7] Additional papers were also added after consultation with domain experts. Nevertheless, even those papers conformed to the inclusion and exclusion criteria.

[7] https://scholar.google.com/.

**Cybersecurity classification.** We classified papers that consider cybersecurity into clusters representing the found research directions based on the keywords used in their title, abstract, and possibly full text if ambiguous. The identified research directions and their keywords are security of Industrial Control Systems (*ICS, SCADA, smart grids*), security of smartphones (*mobile, phone, Android*), web-application security (*web, information system, website*), network traffic security (*DNS, IDS*), attack inspection (*attack inspection, identification, extraction, observation*), outlier user behavior (*outlier, behavior*) and fraud detection (*fraud detection*).

For papers that contain keywords from multiple directions, we classified them into the direction that was considered as the primary of the paper.

**Reliability classification.** For the reliability classification, we used the same principle as with the cybersecurity part of the literature review. The papers were classified based on the keywords used in their title and abstract. If the direction was not explicitly specified, we looked into a broader context of the paper at hand. The identified direction and the keywords used for their classification The identified directions and the keywords used for their classification are the quality assurance (*bug, defect, testing*) and error detection (*failure, error, runtime, monitoring, auditing*).

## 5. Process mining used for cybersecurity

Similarly to the general usage of process mining [10], the majority of the found research cybersecurity papers is using process mining for the analysis of past events. Yet, some papers perform online analysis of real-time events. Some papers are using automatic analysis, i.e., the designed or discovered process models are used for further analysis. On the other hand, some approaches are relying on manual analysis, e.g., a person visually analyzed the discovered model. We found many use cases, mainly about the detection of cyberattacks. In the majority of papers, expert knowledge is applied. Only a few papers do not utilize it in the analysis [23,65].

Table 2 contains all the reviewed research papers. It shows in which research directions the process mining approach is applied, the main target period of the analysis, used type of process mining (process discovery, conformance checking, process enhancement), and whether the proposed technique requires expert knowledge which might affect its usability. It also contains information about the type of model analysis: whether it is performed manually or automatically. Additionally, Table 3 contains the use cases of process mining usage in found papers.

In the following, we discuss the approaches using process mining for cybersecurity in the individual identified research directions.

**Table 2**
Cybersecurity and software reliability papers included in the review.

| Paper | Category | Research direction | Target period | PM Type[a] | Expert knowledge | Model analysis |
|---|---|---|---|---|---|---|
| [54] | Cybersecurity | Security of ICS | Past | PD | ✓ | Automatically |
| [55] | Cybersecurity | Security of ICS | Past | PD & CC | ✓ | Automatically |
| [56] | Cybersecurity | Security of smartphones | Past | PD | ✓ | Automatically |
| [57] | Cybersecurity | Security of smartphones | Past | PD | ✓ | Automatically |
| [58] | Cybersecurity | Security of smartphones | Past | PD & CC | ✓ | Automatically |
| [22] | Cybersecurity | Network traffic security | Past | PD | ✓ | Manually |
| [59] | Cybersecurity | Network traffic security | Past | PD | ✓ | Manually |
| [60] | Cybersecurity | Network traffic security | Present | PD & CC | ✓ | Both |
| [61] | Cybersecurity | Web-application security | Past | PD | ✓ | Automatically |
| [62] | Cybersecurity | Web-application security | Past | PD & CC | ✓ | Automatically |
| [63] | Cybersecurity | Web-application security | Past | PD & PE | ✓ | Manually |
| [24] | Cybersecurity | Web-application security | Past | PD & CC & PE | ✓ | Manually |
| [64] | Cybersecurity | Attack inspection | Past | PD | ✓ | Manually |
| [65] | Cybersecurity | Attack inspection | Past | PD | ✗ | Automatically |
| [66] | Cybersecurity | Attack inspection | Past | PD | ✓ | Manually |
| [67] | Cybersecurity | Outlier user behavior detection | Past | PD & CC | ✓ | Both |
| [9] | Cybersecurity | Outlier user behavior detection | Past | PD | ✓ | Manually |
| [68] | Cybersecurity | Outlier user behavior detection | Present | PD & CC & PE | ✓ | Automatically |
| [69] | Cybersecurity | Outlier user behavior detection | Present | CC | ✓ | Automatically |
| [70] | Cybersecurity | Outlier user behavior detection | Past | CC | ✓ | Automatically |
| [23] | Cybersecurity | Outlier user behavior detection | Past | CC & PE | ✗ | Automatically |
| [71] | Cybersecurity | Fraud detection | Past & present | CC | ✓ | Automatically |
| [72] | Cybersecurity | Fraud detection | Past & present | CC | ✓ | Automatically |
| [73] | Cybersecurity | Fraud detection | Past | PD & CC | ✓ | Automatically |
| [74] | Cybersecurity | Fraud detection | Past | PD | ✓ | Manually |
| [75] | Reliability | Quality assurance | Past | CC | ✓ | Automatically |
| [76] | Reliability | Quality assurance | Past | PE | ✓ | Automatically |
| [77] | Reliability | Quality assurance | Past | PD | ✓ | Manually |
| [25] | Reliability | Error detection | Present | CC | ✓ | Automatically |
| [78] | Reliability | Error detection | Past & present | PD & CC | ✓ | Automatically |
| [79] | Reliability | Error detection | Past | PD | ✓ | Manually |
| [80] | Reliability | Error detection | Past & present | PD & CC | ✓ | Automatically |
| [81] | Reliability | Error detection | Present | CC | ✓ | Automatically |
| [82] | Reliability | Error detection | Past | PD | ✓ | Automatically |
| [83] | Reliability | Error detection | Past | PD | ✓ | Manually |

[a]PD — process discovery, CC — conformance checking, PE — process enhancement.

### 5.1. Security of industrial control systems

Industrial control systems (ICS) are present in critical domains like manufacturing, transportation, and energy sector, where they are responsible for production, monitoring, and control. Naturally, security in these domains is vital as successful attacks could cause significant money loss, physical damage, or injury. The domains are often paired with the term critical infrastructures, emphasizing extensive need for dependability.

Specifically, Bernardi et al. [54] studied the detection of anomalous behavior in energy usage from smart meter readings. They classify the readings based on the expert knowledge into several levels. Then, they use process mining for discovering the behavior of customers over time by looking at how the levels of energy usage were changing. They performed process discovery for several periods with the same length. The output is a sequence of time-evolving graphs over a bigger period. Therefore, in this sequence, they can compare the graphs with each other and find anomalous periods with potential security implications. Two similarity measures were used, i.e., the Hamming distance and cosine similarity [84].

In addition to process discovery, Myers et al. [85] applied the conformance checking method in their work. They firstly investigate the suitability of process-mining discovery algorithms for the detection of cyberattacks in industrial control systems. To this end, they compare five algorithms by the ability to create a usable model, the accuracy and the simplicity of it. Then, with conformance checking, they evaluate the most suitable process discovery algorithm by comparing the number of detected anomalous cases, trace fitness, and time. As a result, they suggest that the Inductive Miner with perfect fitness is the most suitable algorithm in this regard. Based on this paper, in their later work [55], they introduced a method for identifying anomalies in ICS and SCADA device logs. This method is evaluated in a case study that showed that process mining could be successfully used to detect cyberattacks and anomalies using conformance checking in this domain.

Surprisingly, not many work was found in this direction. All identified approaches utilize automatic analysis of past events, but the techniques and use cases widely differ. It is evident that process mining is indeed helpful in this direction. However, the coverage is currently lacking. Future research in this direction could either improve current use cases or extend the usage of process mining for other use cases.

### 5.2. Security of smartphones

Smartphones are ubiquitous devices operating in a unique context close to end-users, with security issues impacting users' privacy (data leaks) or financial implications (credit card theft). The process mining approaches in this regard focus, similarly to ICS direction [54], on malware detection [56,57] by comparing discovered models. In addition, detection specific attacks are approached by conformance checking [58].

The approach for malware detection and phylogeny analysis in smartphone applications was proposed by Bernardi et al. [56]. It applies the process mining on systems call traces to characterize the application behavior. The authors defined Syscalls Execution Fingerprint (SEF), which contains the behavioral model. This behavioral model is a set of declarative constraints between system calls represented by the Declare declarative constraint language [86]. SEF is used as a fingerprint for malware detection, so to detect malware, they compute SEFs of known malware families. These SEFs are then compared to the SEF of the examined application. This approach can recognize malicious behavior based on this comparison. It was evaluated on a dataset of 1200 infected applications across ten malware

**Table 3**
Use cases of papers included in the review.

| Paper | Use case |
|---|---|
| [54] | Integrity attack detection |
| [55] | Detection of cyberattacks |
| [56] | Malware detection |
| [57] | Malware detection |
| [58] | Attack detection |
| [22] | Detection of spam attacks |
| [59] | Alert visualization |
| [60] | Classification of attack traces |
| [61] | Enforcing of security policies |
| [62] | Detection of abnormal behavior on social networks |
| [63] | Detection of attacks and new behavior |
| [24] | Deviation detection in IS audits |
| [64] | Analysis of attack process |
| [65] | Ransomware detection |
| [66] | Attack vector identification |
| [67] | Insider threat detection |
| [9] | Detection of abnormal behavior |
| [68] | Detection of tampered data |
| [69] | Automatic resolving of issues |
| [70] | Security-critical deviations detection |
| [23] | Detection of non-conforming user behavior |
| [71] | Classification of traces |
| [72] | Detection of security breaches and frauds |
| [73] | Process-based fraud detection |
| [74] | Fraud detection |
| [75] | Diagnosing service implementation |
| [76] | Isolating and replicating failure as a test case |
| [77] | Detection of defects in acceptance testing |
| [25] | Runtime verification of behavioral properties |
| [78] | Detecting failures during a rolling upgrade |
| [79] | Identification of bugs in evolving software |
| [80] | Detection of application failures |
| [81] | Monitoring of correct configuration |
| [82] | Monitoring of workflow resilience |
| [83] | Auditing of smart contract executions |

families. It was also proven that it could be used in phylogeny tracking because it could identify variants of the same malware family. They also performed the evaluation on the transformed dataset, where they did current anti-malware obfuscations. Their approach greatly reduced the number of false negatives in this case.

Phylogenic analysis and malware family detection were also performed by Cimino et al. [57]. In this work, process discovery was used to obtain temporal logic formulae which were used in formal model verification. Each path of the discovered process model is transformed into a temporal logic formula. This approach was also evaluated and confirmed to offer an effective solution for this problem in smartphone applications.

The process data from activity logs of Android devices were also used by process mining analysis by Hluchy and Habala [58], who applied conformance checking in addition to process discovery. From the phone logs, they chose OS-generated information about specific performed actions, browser history, and network connection log. Then, they performed an attack in which the user activated a malicious URL, which resulted in downloading personal user data via a known vulnerability, and its discovered model. The used conformance checking technique considers this process model for the offline detection of that attack from examined smartphone logs. This approach raised many false positives, which they concluded is caused because of the simplicity of the attack and the low quality of Android logs.

All the approaches within this direction deal with the automatic analysis of past events. Notably, there is an interesting usage of the declarative approach in two approaches [56,57], which can be attributed to diverse options in smartphone usage. This is further supported by the third work [58], where the focused model covers malicious usage instead of more traditional non-malicious. It is shown that process mining neatly exploits the available data but the number of approaches is surprisingly low given the wide range of available data discussed in the papers.

### 5.3. Network traffic security

In this direction, the focus is put on network traffic data, one of the prime data sources for security analysis. For process mining, there is a dominance of process discovery techniques in found papers. It is usually paired with the manual analysis where the discovered visualized model is analyzed by an assigned expert [22,59]. Notably, one approach performs real-time process discovery for the visualization of a global attack model and conformance checking for the attack classification [60].

Bustos-Jiménez et al. [22] proposed an approach for the detection of unexpected behavior in DNS operations. Their focus is on the detection of a spam attack on DNS servers. From DNS logs, the method builds event logs that are compatible with process mining. They visually compared the discovered model from a dataset with included spam attacks and a dataset without them. Based on this comparison, they were able to identify this type of attack.

Another approach that used visual analysis was proposed by Alvarenga et al. [59], who applied the process mining discovery technique on logs of alerts from Intrusion Detection Systems to create visual models. Complex and large event logs are clustered using hierarchical clustering to provide better user-friendly visualization. This method should help with the manual evaluation of raised alerts from Intrusion Detection Systems. They performed a case study of the proposed approach on an alert dataset from a university.

The approach of Coltelese et al. [60] is the only one in this category that performs analysis in real-time. They aim to filter the amount of IoT attacks that need to be handled by security operators, while at the same time providing them the global attack model that updates automatically using online process discovery. The filtering is done by conformance checking that outputs the fitness of an incoming attack trace with the global attack model. If the fitness value is low, which means that this attack is new, then the attack is sent to the security operator for inspection. In their approach, they assume that as input, they have traces of attacks, so they do not deal with the attack detection.

Prevalently, manual analysis on discovered models is employed in network-traffic security direction. A possible explanation is that the network traffic is complicated, and therefore cognitive skills for the process analysis are required or because the research in this area is still immature. On the other hand, one approach [60] performs automated analysis of the models in real-time, further including conformance checking. This result alone shows the possibilities in the network traffic area and should motivate further research focusing on different attack types and protocols with automation in mind.

### 5.4. Web-application security

This research direction includes publications focusing on the security of information systems, social-network websites, or other web applications, where a security breach could lead to the disruption of services or serious data leaks. We observed a variety of approaches combining the process mining methods, and even split between approaches utilizing automatic [61,62] and manual [24,63] analysis.

In [62], the authors proposed the approach for detecting abnormal user behavior in social network websites to prevent cybercrime. Firstly, they discover a model for normal user behavior using genetic process mining. Then, they identify the abnormal behavior of users by conformance checking. They performed a case study on the Facebook community.

Compagna et al. [61] proposed a tool named *Aegis* that improves the security of web applications by enforcing security policies. This tool uses process mining to discover workflow models of the target application. These models are obtained from sets of user actions that occurred while interacting with web applications. Therefore, there is a need to simulate real users' foreseen behavior first. After the process discovery, the user of this tool can also specify other policies,

like authorization constraints. The model is then transferred into a reachability graph that represents all possible valid executions of the web application workflow. Using this method, a run-time monitor is synthesized. This monitor is used by a proxy between the user and web application. Based on its output, the proxy either forwards the user requests to the application or drop them.

The security of web applications, specifically web information systems, was also considered in work by Bernardi et al. [63], who proposed a method for improving the security of these systems. This approach utilizes process mining and model-driven engineering. First, they specify the system behavior with Unified Modeling Language, from which they automatically generate a formal model. At the same time, they preprocess the obtained logs of the system to get the event logs, which can be used by process mining techniques. To identify deviations, they use ProM [87] visualizations. Those deviations were filtered, and based on the output of the fuzzy mining discovery technique [33], they could be classified as an attack or the new behavior. The classifier, in this case, is the HTTP request code. This method was applied to a web information system for managing the publications.

Similarly to the previous work, Zerbino et al. [24] proposed a process mining methodology in which they manually detect deviations. In this case, it was used for audits information systems, and they used the Disco tool [88]. First, they discovered a process model from historical data. Using Disco, this model can be automatically enriched with other perspectives, like time or organizational perspective. With this tool's visualizations, they manually detected several deviations in a case study. They mentioned process mining advantages over other audit approaches, like better depth of analysis, broader scope, and easier automation.

Web-application security direction contains several approaches, all utilizing process discovery but often combined with other techniques. Remarkably, there is greater utilization of process enhancement [24,63], mainly regarding the time perspective, but only for manual analysis. This direction shows a wide variety of process mining usage for cybersecurity, prompting to utilize the ideas for different domains.

### 5.5. Attack inspection

The direction of attack inspection consists of inspecting how the specific attack is performed, supporting a better understanding and preventing future attacks. In this direction, the found papers only utilize the process discovery technique for the analysis of past events.

Viticchie et al. [64] used process mining for the investigation of the process of the attacks on a small application with different levels of protection, which was performed as an experiment. Every participant filled the report in which their attack strategy was described. These texts were annotated, and process discovery was used on the traces of annotations. The discovered model was used to find out the attack process, the differences between successful and unsuccessful attack processes, and whether this process was influenced by the level of protection used in the application software.

Attacks were also inspected by Macak et al. [66], but a the different point of view. This work is focused on the unintentional insider attack vector identification. Process discovery is used on the event logs produced by the simulation games platform which simulate the working environment for players.

The other approach in this domain is aiming for the inspection and detection of ransomware. Bahrani and Bidgly [65] created event logs from harmless applications and ransomware families. Those logs contained three types of registry events. Then, for each software, they discovered a process model, and from each process model, they extracted the frequencies of each transition. These data then can be used by a classification algorithm to identify ransomware in the application software. Thanks to this, no expert knowledge is required to use this approach.

For the attack inspection, all found approaches analyzed attacks in the past using process discovery. Automatic analysis was performed in one work [65] but without any expert knowledge. This might indicate that it might be hard to transfer the expert knowledge into the automatic system for process-aware analysis of attacks. Future research in this direction could tackle the problem with expert knowledge transfer.

### 5.6. Outlier user behavior detection

The outlier user behavior detection is motivated by finding suspicious user activities, which could be malicious in nature. The found approaches can be divided into two main categories: approaches that take advantage of expert knowledge [9,67–69], and approaches that do not need expert knowledge [23]. In those research approaches that do not incorporate expert knowledge, we can see two main methods. Either they are focused on security as their primary goal [23], or their focus is to filter outliers for a better process model [89,90]. Note that the latter is not included in our literature review as their primary focus is not cybersecurity.

The exploration of process mining potential for security was performed in multiple approaches. Genga and Zannone [9] designed a methodology for behavior analysis using process mining and applied it to a real event log, while Macak et al. [67] focused on the process mining behavior analysis of insiders in organizations. Both approaches present a set of challenges of a process mining application for security based on their experience, e.g., dealing with the data collection, preprocessing, and selection of the proper features and technique for the analysis.

Li et al. [68] proposed an online token-based monitoring framework for process interaction between collaborative sub-processes in different departments. Based on the business requirements, it ensures the process security by detecting the outer tampering of the data, for example, through the Internet. In this framework, they use a strategy based on the pre-checking of inputs and the post-checking of outputs of tasks. To mine a global interaction process model, they use Interorganizational Workflow, which is based on Workflow net. It can represent a global workflow process that includes local workflow processes and an interaction structure. To discover local processes, they use $\tau$ algorithm, but for mining the interaction places between them, they had to adjust this algorithm and named it $\tau^*$. In the end, they combine local processes through interaction places. The monitoring and detection of the undesired behavior is based on the token-replay conformance checking method.

A conformance checking technique was also employed in a proposed method by Salnitri et al. [70] to identify the security deviations in process executions. This approach is implemented in the loan management process of a financial institute. It aims to automatically analyze process executions and identify deviations from a previously defined process model using conformance checking. Then, it determines which deviations are security-critical, based on predefined security policies, and visualize them. For the definition of business processes and security policies, they use the SecBPMN2 modeling language. However, to realize the conformance checking step, they convert the process model to the Petri net notation. After this step, the discovered deviations are transformed back to the SecBPMN2 modeling language. Their approach is fully supported by an extended version of the STS-Tool, which is a software that helps in maintaining a high level of security in socio-technical systems [91].

A system for online analysis was also presented by Talamo and Arcieri [69]. It is used for the operational support of distributed business processes, which handle sensitive data and require a high level of security, with real-time compliance checking. This system uses Validation Trees to process the data, detect anomalies, and create reports. The automated process validation and troubleshooting functionality are integrated with IT service desk operations. So, in this case, the security is not improved by detecting undesired behavior, but by reducing the

human intervention in Service Desk because of automatic resolving of user issues.

There are several papers that are detecting the outliers in the processes. However, in this literature review, we exclude those that do not have the security as their primary goal in their proposed approach. Alizadeh et al. [23] proposed an auditing approach that combines data and process perspectives to detect non-conforming user behavior with conformance checking. This approach can identify the previously undiscovered deviations.

In this direction, conformance checking is prevalent, emphasizing its effectiveness in outlier behavior detection. Additionally, all approaches except one [23] use expert knowledge for the analysis. Remarkably, there is a wide variety of use cases and techniques, despite this research direction being the largest one found. We speculate that the cause is that user behavior can be very unpredictable, forming a complex process, but at the same time, it can be very harmful in cybersecurity. This variability indicates a potential for future research as the methods are not stale yet.

### 5.7. Fraud detection

Fraud detection is a specific research direction aiming to detect false pretenses of entities. Typical examples are unusual processes, violating a rule or policy. The found approaches utilizing process mining in the cybersecurity context tend to employ conformance checking with expert knowledge.

Fazzinga et al. [92] proposed a method for online and offline classification of event log traces as potential security breaches. They create a security breach model, which is used later in conformance checking. In their work, they are trying to solve the problem with the mapping between high-level and low-level operations. It is important in this scenario because they claim that security breach models are typically described as high-level activities, but log traces are typically performed as low-level operations. They used a probabilistic approach in the created model and the following conformance checking. In their following paper [93], they proposed a classification framework that combines their previous work [92], a model-driven method with an example-driven classification. In a model-driven approach, a security breach model is created, and incoming traces are classified based on conformance checking. In the example-driven approach, a set of previously labeled traces is used for later classification. This approach was experimentally validated in their next paper [71].

Security breaches were also the aim of the work of Böhmer and Rinderle-Ma [72], who proposed an anomaly detection strategy in process execution events to prevent not only security breaches but also frauds. They include the control flow, time, and resource perspective into one anomaly detection approach. They try to detect point, contextual, and collective anomalies. They also try dealing with unexpected events that might not indicate an anomaly. They propose to construct a likelihood graph, which represents the likelihood of the expected execution of events. Firstly, they create a basic likelihood graph with activities and their probabilities. Then they extend it with other perspectives. For detecting the anomaly, they compare the likelihood of the currently executing event with the likelihood of recorded comparable events based on the likelihood graph.

Baader and Krcmar [74] were also using process mining for fraud detection. Specifically, they present a method for reducing the number of false positives in this detection. They combine the red-flag approach and process mining for identification and visualization of possible undesired process instances. Potential frauds are identified and then visualized with the fuzzy miner. Their approach was compared to two other approaches, and they got a lower number of false positives. However, they detected over half fewer frauds than one chosen variant. They discuss that process mining gives several other advantages to a classical red flag approach, such as easier dashboard analysis and visualization.

In particular, Huda et al. [73] proposed a method for the identification of process-based frauds in a credit application. After discovering a process model, they perform conformance checking analysis to check how many events were skipped. Additionally, they also perform analyses from different angles, like performance, segregation of duty, and role analysis. Furthermore, they proposed ten attributes that can be used as an output of the log analysis. Based on the occurrences of violations in these attributes, the type of fraud can be obtained.

In this research direction, there is a clear motivation for easing the manual analysis and automation. Indeed, all approaches but one employed automated conformance checking. The remaining approach introduced a method for reducing the number of detected false positives, making it easier for the person to analyze the rest manually [74]. Remarkably, the prevalent approach is the creation of a custom algorithm which is then used in the analysis itself. This fact might be caused by the specificity and variability of frauds that might exist, encouraging a further investigation in the area.

## 6. Process mining used for software reliability

The most significant observation regarding the use of process mining for software reliability is a notable sparsity of research papers. It is despite the fact that the field was pointed out as appealing to study further [94]. The literature review also found an interesting trend concerning reliability-focused studies. The reliability is more frequently approached in processes involving humans [95], than critical software.

Regarding the use cases of the reviewed papers, it is interesting that the majority of them focused on finding bugs or errors in general and not explicitly tackling the reliability of software systems. Also, the models are primarily analyzed automatically. The reason for this approach is partly because the process mining techniques in the reviewed papers were part of a more sophisticated method or framework [25,75].

In the classification of the papers, two main directions can be spotted, which divides them roughly in accordance with the software lifecycle, i.e. pre- and post-deployment. The dominant class is focusing on activities after the deployment like monitoring or maintenance. In comparison, the less frequent one is aiming to support quality assurance before the software is rolled out. Papers aimed at reliability in different contexts are very often focused on the human side of the issue, not software reliability as such, so they are not included in our literature review.

Table 2 contains all the identified research papers. It shows the direction in which the process mining is situated, the used process mining technique (process discovery, conformance checking, process enhancement), the main target period of the analysis, and whether the proposed technique needs expert knowledge. It also contains information about the type of model analysis: whether it is performed manually or automatically.

### 6.1. Error detection

The direction focusing on error detection deals with identifying improper behavior and finding its root cause within the part of a software system in question. Specifically, non-malicious and non-deliberate faults are emphasized.

Error detection is the most prevalent domain when it comes to the application of process mining for software reliability. The prevalence might be caused by the fact that in the case of software reliability, authors seem to focus on generic methods and frameworks [80,82], without strict application-domain focus. On the other hand, some trends like service-oriented architectures [81], adaptive middleware [25], blockchain [83], or cloud [78], can be seen. Generally, the error detection direction is concerned with runtime behavior analysis to detect bugs in the implementation [25] or assisting the deployment [78], maintenance [79], and monitoring [82].

Service orientation is a topic of interest by van der Werf and Verbeek [81], who present the application of process mining for continuous auditing of service behavior. The aim is to monitor violations of security requirements in support of configuration management. To this end, the authors present a tool that analyzes event logs from executions across the service landscape using semantic process mining, which combines process mining with semantic web techniques. A key concept is enabling relations between elements in the event log and other elements using annotation rules. The tool utilizes process mining to check conformance to defined constants. Concretely, reliability and security-related constraints are discussed.

Software design and runtime behavior, albeit in the case of adaptive middleware, is considered by Rosa et al. [25], who presents a solution for its implementation and execution. Besides covering the basic requirements of adaptive middleware systems, the authors include techniques for verification of their implementation. The process mining is used for runtime verification of behavioral properties of an implemented adaptive middleware system. The behavior is verified by conformance checking against its specification in LTL. In the case of undesired behavior, the proposed solution automatically creates and executes an adaptation that mitigates the issue. Thus, making the system more robust.

Detection of application failures is the topic of highly detailed work by Pecchia et al. [80]. The proposed approach consists of multiple steps that utilize different methods of process mining. The main ingredient of the approach is the application logs. Those are firstly used to construct a reference model capturing normal behavior using process discovery. Subsequently, the logs are used, possibly in real-time, for conformance checking against the reference model to discover failures in the application. The work is highly detailed in both implementation and evaluation. Furthermore, it considers the presence of noise in application logs.

Applications of process mining in the field of software maintenance are explored by Gupta [79]. The author presents four applications of process mining assisting in improving various processes. Concretely, management of software projects, ticket resolution, and software bug detection. The latter is highly relevant for increasing the overall reliability of the software at hand. In this case, two models are obtained by performing process-discovery algorithms based on execution logs — the current stable and the new software versions. The models are then compared for differences. However, the paper does not go into much detail about the actual technique used and serves more as an outline for future work.

The work by Xu et al. [78] presents an error detection method aimed at sporadic operations like software deployment. The specific use case presented in the paper is a rolling upgrade in a cloud environment. In order to achieve accurate detection, the method first creates a process model of the operation from regular, un-anomalous logs using process discovery. Afterward, the conformance checking is used as part of the analysis to determine if the process is running correctly. Here, process mining is just one step in a more sophisticated analysis.

Resilience monitoring of executable business processes, with the focus on time, is discussed by Zahoransky et al. [82]. The authors present a mathematical framework to define the notion of resilience in workflows. The main idea is to automatically extract the probability distribution of time characteristics of a given workflow. Process mining is used to obtain the time information from an event log, which is then used to calculate the probability distributions. The approach can be utilized to predict and monitor the resilience indicators of the executed workflow, possibly enacting timely countermeasures if the levels drop.

Auditing, specifically aimed at blockchain smart contracts, is a focus of the work by Corradini et al. [83]. The authors devised a method that extracts the transaction logs, cluster them by the sender to create an event log in XES format, discovers a model, and analyzes it. Process mining, concretely the process discovery technique, is used in the third phase to generate three candidate models using different algorithms. Out of the models, only one is chosen based on the quality characteristics. The resulting model is then used in the last step to find discrepancies with the specification. The conformance is assessed manually, with the hints to use model checking techniques.

In summary, the process mining techniques utilized for error detection focus on either analyzing the runtime behavior and providing assistance during operations. Overall, the trend inclines towards automation, which seems to be the primary motivation apart from taking advantage of process events. In this regard, comparing previous and current behavior is a very appealing usage of process mining employed by [78,80]. Another trend is to use process mining to continuously check defined nominal behavior, as used by [25,81,82]. Still, the coverage of application domains, leaving many possibilities for future work. Furthermore, while several approaches utilized artifacts from operations, the area is not fully explored, for instance, cloud provisioning or continuous integration. Possible process mining utilization in software operations is provided by [79], but most presented cases are out of the scope for this review.

### 6.2. Quality assurance

The topics in the quality assurance direction could be summarized as aiding the developers with software reliability before deployment. It focuses on identifying issues before the software is put into production. In this direction, both direct (validation and verification) and indirect (QA process) support is relevant. Papers within this typically aim at giving hints and pinpointing bugs [77] or outright generation of test cases [76]. The human-centric development aids, like [96] focusing on agile development, were only considered if the software product itself was part of the consideration.

Service-oriented design, namely the service substitution and its behavior, are explored by Sahl and van der Aalst [75]. The authors present a formal model that is describing the service behavior, as well as formal methods and an algorithm that enables correct substitution. The correctness is checked in both design-time and runtime. Process mining, more precisely conformance checking, is utilized in the former and is applied to ensure that the implementation of a service conforms to a contract and modeled behavior. Therefore, process mining aiding in bug detection and, as a consequence making the service implementation more reliable. It is important to note that the method is focusing solely on functional behavior, not considering the non-functional one.

The practically oriented work by Rubin et al. [77] outlines the possibilities of process mining applied to software projects. Here, the authors demonstrated two distinct use cases focusing on the analysis of software system using process discovery. In the context of this review, the first case is particularly interesting as it aims at detecting software defects during acceptance testing. The discovered model based on the software system logs is presented as a way for developers to pinpoint bugs from debugging and exception logs by manual review.

Process mining principles are used by Lübke [76] for generating encapsulated bundles to contain and replicate a failure in a business process management system, outside of the production environment. The bundles, called Replication Test Cases, contain an isolated unit test that enables the reproduction of the failure without any dependencies. Therefore, developers can fix the issue more efficiently. For the generation of the bundles, the author proposed a specialized algorithm based on a process mining approach that utilizes both event logs and the reference process model.

A common trend in quality-assurance research direction utilizing process mining is to either minimize the introduction of bugs into software [75] or to diagnose them [76,77]. While quality assurance consists of many processes generating artifacts, they are not explored from a process mining perspective. For example, code reviews or analyzing changes in the codebase leaves space for future work. Furthermore, providing the developers with various process models could assist in software understanding and diagnosis, as proposed by [77].

## 7. Results

This section formulates the insights from the literature review into the answers to our research questions.

### 7.1. [RQ1] What are the research directions in which the process mining is used to ensure cybersecurity and software reliability?

Within our review, we have identified 35 approaches employing process mining in cybersecurity or software reliability, which we have structured according to the direction of the research problem they are addressing. These include the security of industrial control systems, security of smartphone devices, web-application security, network traffic security, attack inspection, outlier user behavior, fraud detection, error detection, and quality assurance. Although this division is by no means perfect, it gives a useful understanding at what research problems the process mining is directed in terms of cybersecurity and software reliability.

In the case of cybersecurity, the most popular research approaches (with the highest number of publications) were targeting general direction towards either detection of outlier user behavior or frauds. The next most popular direction was the cybersecurity of network elements, namely focusing on the applications related to networking in connection with websites, information systems, and other technologies that primarily communicate through networks. The research in the remaining directions is rather sparse.

The most popular research direction in the case of software reliability is the error detection. In this case, process mining is typically utilized to monitor runtime behavior in search of errors or abnormal behavior. Curiously, in the quality assurance research direction, the utilization is very similar, however focusing on aiding the development of the systems, rather than its operation and maintenance. Indeed, some outlier approaches can be found, like generating test cases. Overall, in software reliability, process mining is primarily used in a very narrow sense to detect faults, errors, and failures. The usage for analysis of reliability attributes is, barring one paper, unexplored.

Surprisingly, we did not discover a holistic approach combining both fields, as stressed by [2]. However, the work by van der Werf and Verbeek [81] contains features of both fields. For the rest of the papers, there are some significant similarities where researchers took analogous approaches across the fields. The most significant overlap lies in detecting various anomalous or undesired behavior, which shares a similar impact while having different root causes. Indeed, utilizing process mining to analyze behavior is a major strength of process mining. Additionally, process mining is utilized for inspection and diagnosis across the fields, focusing on attacks and errors, respectively. This hints towards similar techniques for general faults regardless of cause.

On the other hand, we noticed some differences regarding the use of process mining between the two fields. First and foremost, cybersecurity seems to be more popular with more than twice the number of papers compared to software reliability. Furthermore, we found out that the application domains (e.g., smart grids, mobile devices, financial systems) in the cybersecurity field are more diverse. In contrast, the distinguishing factor within software reliability is technology or platform.

Domains that are generally dominant in process mining fields in the general process mining realm were surprisingly not found among the results. Those domains include healthcare, manufacturing, education, finance, and logistics. Thus, we assume that in these domains, even though they are generally popular, the process mining techniques have not yet been properly explored for the purpose of cybersecurity or software reliability. Alternatively, they could be employing general domain non-specific techniques and thus not published mentioning a particular domain.

An exception to the claim is a paper, removed from the results due to the exclusion criterion on full paper. The work is dealing with software reliability in the healthcare domain. It is a short, workshop paper presenting potential filed and future research focus [97].

An interesting fact that was discovered is that process mining had been extensively used to improve safety-critical processes involving humans [98,99]. However, the domain-specific computer or software involvement remains rather unexplored. At least using the process mining techniques.

### 7.2. [RQ2] Which process mining techniques and approaches are utilized to ensure cybersecurity and software reliability?

Based on the discovered papers, most research approaches are targeting the past period for analysis. The papers utilizing real-time analysis are mostly limited to directions explicitly focusing on detection (i.e., outlier user behavior detection, fraud detection, and error detection) [25,69,72,81]. Nevertheless, we noticed a trend that many papers, e.g., [75], targeting the past mention the possibility of real-time application, which is, however, not demonstrated or left for future work. Regarding the real-time analysis, only one paper that utilized online process discovery was found [60]. The rest of the approaches that analyzed data in real-time used online conformance checking [68,69].

Several patterns regarding the used types of process mining were identified. Process discovery and conformance checking techniques are quite common in these areas. On the other hand, process enhancement was used rarely [63,76]. An interesting trend is that several papers used the combination of process discovery and conformance checking in their approaches [58,73]. A typical scenario is to generate a reference model using process discovery and then apply conformance checking for new cases. Similarly, some approaches described custom conformance methods [77,79]. In such a case, they often use process discovery to obtain inputs for this custom method. Otherwise, the research approaches mainly just use either process discovery [54] or conformance checking [70]. The latter often conformance checking for real-time verification against ground truth [25,81].

In many papers, expert knowledge is needed in order to analyze the data. Yet, there are exceptions, like a paper [65] that used process mining for ransomware classification in application software and papers targeting cybersecurity in the business processes domain that is aimed at outlier detection, without any need for expert knowledge [23].

Regarding the analysis, the majority of the papers analyze the obtained model automatically, partly because the process mining technique is very often used as a part of a more sophisticated method [57,65]. It seems as automation is a motivation of numerous approaches [25,60,78]. However, there exist papers that utilize only manual analysis based on the discovered model. In this case, the goal is to employ visual analytics to discover unexpected anomalies [22,59], or the paper [83] that presents an early stage of research where the analysis is not yet automated.

Specific techniques used in found approaches are in Table 4. From all 35 found approaches, seven papers do not specify all process mining techniques they use and six use their own custom algorithm. Only three from the specified techniques used declarative process mining approach.

The most popular technique for process discovery is fuzzy miner [33]. We assume that the reason is that it is configurable technique that can deal with unstructured processes and generate simplified process models, which is very desirable in cybersecurity and software reliability. However, similar technique with same advantages, heuristic miner [32], is not so popular. We think it is because of user-friendly analytical tools that support process discovery using fuzzy algorithm, e.g., Disco (which was used in the majority of found approaches utilizing fuzzy miner). Some newer approaches used Python library PM4Py [100], created in 2019, which might indicate more

**Table 4**
Techniques used in papers included in the review.

| Paper | Techniques |
|---|---|
| [54] | Fuzzy miner (PD) |
| [55] | Inductive miner (PD), A*-based (CC) |
| [56] | Declare miner (PD) |
| [57] | Fuzzy miner (PD) |
| [58] | Inductive miner (PD), A*-based (CC) |
| [22] | Fuzzy miner (PD) |
| [59] | Fuzzy miner (PD) |
| [60] | Online heuristic miner (PD), alignment-based (CC) |
| [61] | Alpha miner (PD) |
| [62] | Genetic process mining (PD), not specified (CC) |
| [63] | Fuzzy miner (PD) |
| [24] | Fuzzy miner (PD), manual (CC) |
| [64] | Fuzzy miner (PD) |
| [65] | Fuzzy miner (PD) |
| [66] | Fuzzy miner (PD) |
| [67] | Heuristic miner (PD), fuzzy miner (PD), LTL checker (CC) |
| [9] | Fuzzy miner (PD) |
| [68] | $\tau$ algorithm (PD), custom algorithm (PD), not specified (CC) |
| [69] | Not specified |
| [70] | Alignment-based (CC) |
| [23] | Custom |
| [71] | Custom |
| [72] | Custom |
| [73] | Not specified |
| [74] | Fuzzy miner (PD) |
| [75] | Alignment-based (CC) |
| [76] | Custom (PE) |
| [77] | Fuzzy miner (PD) |
| [25] | LTL checker (CC) |
| [78] | Fuzzy miner (PD), token-based (CC) |
| [79] | Not specified |
| [80] | Fuzzy miner (PD), token-based (CC) |
| [81] | Fuzzy miner (PD) |
| [82] | Not specified |
| [83] | Heuristic miner (PD), inductive miner (PD), split miner (PD) |

popular usage of this library in the future, as it supports multiple process mining algorithms and is not so limited as analytical tools.

Regarding the conformance checking, we did not find any prevalent technique. Often they were not specified or were customly created for the given task. The most used techniques were alignment-based [37].

### *7.3. [RQ3] What are the current gaps and possible research directions in the usage of process mining for cybersecurity and software reliability?*

Research gaps and possible research directions can be observed in multiple aspects. First, there is a gap in the usage of process mining in cybersecurity and software reliability in several domains that otherwise do utilize process mining (for other applications), like healthcare, manufacturing, education, and logistics. Although we found some research papers focused on these areas, they were dealing with safety-critical processes involving humans and not computer systems [98,99], or just presented upcoming research direction [97]. Secondly, we detected a substantial untapped potential in the application of process mining within a larger context of cybersecurity and software reliability. Areas such as availability, robustness, or resilience are rather unexplored with respect to process mining, although they are implied in literature or applied in a non-IT field. In summary, and based on the found papers, we see a great potential of process mining for cybersecurity and software reliability. Daunting future directions could utilize the good practice from non-IT areas, where faults, anomalies, safety, or robustness of a process is evaluated.

Furthermore, we want to emphasize a set of criteria required to approach cybersecurity and software reliability problems in a process-aware manner. We believe this will help other researchers discover new use cases for process mining. These criteria are:

- Events can be ordered into a sequence, and membership of the event in the sequence can be determined.
- The beginning and end of the process need to be clearly definable.
- The perspective of the process needs to be established. The process can be analyzed either from the perspective of entities that perform actions or from the perspective of objects that are being used by these entities.
- When it is needed to analyze the process of groups of entities, such entities need to have the same expected behavior.
- Events that can happen in the process should be categorical to get a reasonable process model. However, numerical data can still be used when they can be easily converted to categorical types.

Other possible research directions stem from how the process mining analysis is performed in the found papers, their motivation, and goals. For example, a frequent topic is a real-time analysis, where process mining is an ideal candidate. However, the existing approaches are almost always static and based on past events. Future work might improve this and focus more on real-time events, allowing for timely insights into possible cybersecurity and software reliability issues. Frequently also, the application of process mining is motivated by automation or easing the manual analytical work. In this sense, we see a strong potential in automated detection and evaluation of non-standard, malicious, or faulting behavior of computer systems.

Finally, as the number of publications is very sparse, there are likely many uninvestigated use cases of process mining in cybersecurity and software reliability. Some of the use cases are pointed out in the discussion of respective research directions. Prime examples include analysis of software changes and behavior, as well as supporting the diagnosis of software issues (both malicious and non-malicious). It may be especially worthy of considering the process mining in critical infrastructures, as they need to be highly reliable and have a strong defense against cyberattacks, which might take advantage of seemingly minor vulnerabilities, yet combined into malicious processes. The process mining might be valuable in the analysis and diagnostics of already running or legacy systems. Specifically, the streaming-based process mining techniques [101,102] can be used in this regard.

## 8. Threats to validity

For the evaluation of threats to validity, we followed the strategy by Zhou et al. [103] for systematic literature reviews. In this section, we assess the potential threats to validity and discuss the precautions that we used to mitigate them through our review. Concretely, the threats are grouped into four categories, i.e., construct, internal, external, and conclusion validity.

### *8.1. Construct validity threats*

We chose six popular digital libraries that are believed to cover the majority of high-quality publications to establish a base collection of papers. To obtain as many papers as possible, we also used snowballing to reduce the possibility of missing relevant approaches. Furthermore, we divided the search and reasoning into two parts because the communities of cybersecurity and software reliability might differ. Regardless of the division, the results of the searches have been combined and deduplicated for subsequent stages.

Additionally, there is a possibility that our defined inclusion and exclusion criteria might have caused that some relevant papers were not included in our review. We tried to avoid this by consulting the set of reviewed papers with multiple researchers in the process mining field.

### *8.2. Internal validity threats*

We divided the filtering of papers into multiple phases. Each phase was performed by a different researcher. In case of any doubts, the paper is marked and discussed with two more researchers to reach the consensus. To further avoid bias in filtering, a sample of 10%

papers filtered out in the first phase was double-checked by another researcher. In this review, we formulated a strategy to guide the review, which is based on the existing guideline for systematic literature reviews [53]. In the reliability search, we included all popular terms which are connected to the reliability and software dependability.

### 8.3. External validity threats

The search is limited to the up-to-date papers over the last six years, from 2014 to 2020. The primary motivation for the restriction to this period is to focus on the most recent research and applications. Secondarily, we expected the more mature approaches among these recent publications. Lastly, the lower bound, the year 2014, also corresponds to a local peak in the number of publications within the Dimensions[8] dataset, showing a slight change of trend.

To evaluate the number of potentially missed publications from the filtered period, we performed an additional search with the same parameters apart from the period. Based on the same duplicity and relevancy ratio assumption, we approximated the filtering of 14 relevant articles published before 2014.

### 8.4. Conclusion validity threats

To ensure the correctness of the extracted data, we formulated the strategy of this literature review and established categories of interest which had to be described for each relevant research paper. The popularity of identified domains was also cross-checked with more general reviews on process mining. Moreover, the interpretation of the data was extensively discussed. Despite our systematic approach, some trends, patterns, and research gaps could have been missed in the review. On the other hand, we believe that the value of the provided summary in this work is not primarily in the research gaps, but in overview of the existing body of knowledge in applying process mining in the area of cybersecurity and software reliability. In this way, we believe we can facilitate the understanding of what attempts exist so that it is easier for the reader to see where they can build on what is existing and where they need to start building their approach from scratch.

## 9. Conclusion

In this paper, we conducted a systematic literature review to provide an overview of research papers that use process mining for cybersecurity and software reliability. In accordance with our initial assumption, process mining techniques have been used for this purpose. Papers summarized in this work show that original research advancements of mentioned directions are possible thanks to process mining. However, the coverage is still rather limited, with numerous research gaps, especially in software reliability. As such, there is much potential in both cybersecurity and software reliability applications of process mining. We identified nine major research directions, discussed how they fit in the overall landscape and presented how they utilize the process mining for these purposes. While we found that the process mining research directions for cybersecurity and software reliability are taking quite different focus, there are similarities in approaches and usage of process mining techniques. For example, a detection of anomalies follows a similar approach regardless of their cause (e.g., errors, frauds). Most importantly, they contribute to the dependability of the systems, which must consider both areas. We demonstrated the feasibility of using process mining with this goal.

Based on this systematic literature review, we pointed out a set of possible process mining research directions that can be taken to tackle the state-of-the-art challenges in cybersecurity and software reliability. Primarily, we would encourage the research community to deeper investigate the domain of critical information infrastructures, which might benefit significantly from more advanced cybersecurity and software reliability techniques. The real-time analysis of systems has a strong potential to utilize the advantages of process mining techniques. Furthermore, it would be indeed beneficial in the advanced detection and prevention of cyberattacks and incidents, enhancing it with a process-oriented approach.

[8] https://app.dimensions.ai/.

## Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

## Acknowledgment

This research was supported by ERDF “CyberSecurity, CyberCrime and Critical Information Infrastructures Center of Excellence” (No. CZ.02.1.01/0.0/0.0/16_019/0000822).

## References

[1] Avizienis A, Laprie J-C, Randell B, Landwehr C. Basic concepts and taxonomy of dependable and secure computing. IEEE Trans Dependable Secure Comput 2004;1(1):11–33. http://dx.doi.org/10.1109/TDSC.2004.2.

[2] Serpanos D. There is no safety without security and dependability. Computer 2019;52(6):78–81. http://dx.doi.org/10.1109/MC.2019.2903360.

[3] Asghar MR, Hu Q, Zeadally S. Cybersecurity in industrial control systems: Issues, technologies, and challenges. Comput Netw 2019;165:106946. http://dx.doi.org/10.1016/j.comnet.2019.106946.

[4] Leander B, Causevic A, Hansson H. Cybersecurity challenges in large industrial iot systems. In: 2019 24th ieee international conference on emerging technologies and factory automation (etfa). 2019, p. 1035–42.

[5] Liu L, De Vel O, Han Q, Zhang J, Xiang Y. Detecting and preventing cyber insider threats: A survey. IEEE Commun Surv Tutor 2018;20(2):1397–417.

[6] Yen T-F, Oprea A, Onarlioglu K, Leetham T, Robertson W, Juels A, Kirda E. Beehive: Large-scale log analysis for detecting suspicious activity in enterprise networks. In: Proceedings of the 29th annual computer security applications conference. Acsac ’13, New York, NY, USA: Association for Computing Machinery; 2013, p. 199–208. http://dx.doi.org/10.1145/2523649.2523670.

[7] Young WT, Goldberg HG, Memory A, Sartain JF, Senator TE. Use of domain knowledge to detect insider threats in computer activities. In: 2013 ieee security and privacy workshops. 2013, p. 60–7.

[8] Senator TE, Goldberg HG, Memory A, Young WT, Rees B, Pierce R, et al. Detecting insider threats in a real corporate database of computer usage activity. In *Proceedings of the 19th acm sigkdd international conference on knowledge discovery and data mining*, 2013, p. 1393–401.

[9] Genga L, Zannone N. Towards a systematic process-aware behavioral analysis for security. In: Proceedings of the 15th international joint conference on e-business and telecommunications - volume 1: bass. SciTePress, INSTICC; 2018, p. 460–9. http://dx.doi.org/10.5220/0006944604600469.

[10] van der Aalst W. Process mining: data science in action. 2nd ed.. Springer Publishing Company, Incorporated; 2016.

[11] v. d. Aalst W. Using process mining to bridge the gap between bi and bpm. Computer 2011;44(12):77–80.

[12] Rozinat A, Günther CW. The added value of process mining. BPM Everywhere: Internet Things Process Everything 2019.

[13] van Genuchten M, Mans R, Reijers H, Wismeijer D. Is your upgrade worth it? Process mining can tell. IEEE Softw 2014;31(5):94–100.

[14] Ghasemi M, Amyot D. From event logs to goals: a systematic literature review of goal-oriented process mining. Requir Eng 2020;25(1):67–93.

[15] Mardani S, Shahriari HR. A new method for occupational fraud detection in process aware information systems. In: 10th international isc conference on information security and cryptology, iscisc 2013, Yazd, Iran, August 29-30, 2013. 2013, p. 1–5.

[16] Geyer-Klingeberg J, Nakladal J, Baldauf F, Veit F. Process mining and robotic process automation: A perfect match. In *Proceedings of the dissertation award, demonstration, and industrial track at bpm 2018 co-located with 16th international conference on business process management (bpm 2018), Sydney, Australia, September 9-14, 2018*, 2018, p. 124–31.

[17] Macak M, Kruzelova D, Chren S, Buhnova B. Using process mining for git log analysis of projects in a software development course. Educ Inf Technol 2021;1–31.

[18] dos Santos Garcia C, Meincheim A, Junior ERF, Dallagassa MR, Sato DMV, Carvalho DR, et al. Process mining techniques and applications - A systematic mapping study. Expert Syst Appl 2019;133:260–95. http://dx.doi.org/10.1016/j.eswa.2019.05.003.

[19] Reinkemeyer L. Process mining in action: principles, use cases and outlook. Springer Nature; 2020.

[20] Keith B, Vega V. Process mining applications in software engineering. In: International conference on software process improvement. Springer; 2016, p. 47–56.

[21] Elkoumy G, Fahrenkrog-Petersen SA, Sani MF, Koschmider A, Mannhardt F, Von Voigt SNn, Rafiei M, Waldthausen LV. Privacy and confidentiality in process mining: Threats and research challenges. ACM Trans Manage Inf Syst 2021;13(1).

[22] Bustos-Jiménez J, Saint-Pierre C, Graves A. Applying process mining techniques to dns traces analysis. In: 2014 33rd international conference of the chilean computer science society (sccc). 2014, p. 12–6. http://dx.doi.org/10.1109/SCCC.2014.9.

[23] Alizadeh M, Lu X, Fahland D, Zannone N, van der Aalst WM. Linking data and process perspectives for conformance analysis. Comput Secur 2018;73:172–93. http://dx.doi.org/10.1016/j.cose.2017.10.010.

[24] Zerbino P, Aloini D, Dulmin R, Mininno V. Process-mining-enabled audit of information systems: Methodology and an application. Expert Syst Appl 2018;110:80–92. http://dx.doi.org/10.1016/j.eswa.2018.05.030.

[25] Rosa NS, Campos GM, Cavalcanti DJ. Lightweight formalisation of adaptive middleware. J Syst Archit 2019;97:54–64. http://dx.doi.org/10.1016/j.sysarc.2018.12.002.

[26] Macak M, Daubner L, Fani Sani M, Buhnova B. Cybersecurity analysis via process mining: A systematic literature review. In: Advanced data mining and applications. Springer International Publishing; 2021.

[27] Cook JE, Wolf AL. Automating process discovery through event-data analysis. In: Proceedings of the 17th international conference on software engineering. Icse '95, New York, NY, USA: Association for Computing Machinery; 1995, p. 73–82. http://dx.doi.org/10.1145/225014.225021.

[28] Datta A. Automating the discovery of as-is business process models: Probabilistic and algorithmic approaches. Inf Syst Res 1998;9(3):275–301.

[29] Agrawal R, Gunopulos D, Leymann F. Mining process models from workflow logs. In: Schek H-J, Alonso G, Saltor F, Ramos I, editors. Advances in database technology — edbt'98. Berlin, Heidelberg: Springer Berlin Heidelberg; 1998, p. 467–83.

[30] van der Aalst W, Weijters T, Maruster L. Workflow mining: Discovering process models from event logs. IEEE Trans Knowl Data Eng 2004;16(9):1128–42.

[31] van Dongen BF, De Medeiros AA, Wen L. Process mining: Overview and outlook of petri net discovery algorithms. In: Transactions on petri nets and other models of concurrency ii. Springer; 2009, p. 225–42.

[32] Weijters A, van der Aalst WM, De Medeiros AA. Process mining with the heuristics miner-algorithm. Tech. rep. wp 166, Technische Universiteit Eindhoven; 2006, p. 1–34.

[33] Günther CW, van der Aalst WMP. Fuzzy mining – adaptive process simplification based on multi-perspective metrics. In: Alonso G, Dadam P, Rosemann M, editors. Business process management. Berlin, Heidelberg: Springer Berlin Heidelberg; 2007, p. 328–43.

[34] Lamma E, Mello P, Montali M, Riguzzi F, Storari S. Inducing declarative logic-based models from labeled traces. In: Alonso G, Dadam P, Rosemann M, editors. Business process management. Berlin, Heidelberg: Springer Berlin Heidelberg; 2007, p. 344–59.

[35] Carmona J, van Dongen B, Solti A, Weidlich M. Conformance checking. Springer; 2018.

[36] Rozinat A, van der Aalst WM. Conformance checking of processes based on monitoring real behavior. Inf Syst 2008;33(1):64–95.

[37] van der Aalst W, Adriansyah A, van Dongen B. Replaying history on process models for conformance checking and performance analysis. Wiley Interdiscip Rev: Data Min Knowl Discov 2012;2(2):182–92.

[38] Fahland D, van der Aalst WM. Model repair—aligning process models to reality. Inf Syst 2015;47:220–43.

[39] Burattin A, Sperduti A, Veluscek M. Business models enhancement through discovery of roles. In: CIDM. 2013, p. 103–10.

[40] Jaisook P, Premchaiswadi W. Time performance analysis of medical treatment processes by using disco. In: 2015 13th international conference on ict and knowledge engineering (ict & knowledge engineering 2015). IEEE; 2015, p. 110–5.

[41] Kurniati AP, Johnson O, Hogg D, Hall G. Process mining in oncology: A literature review. In: 2016 6th international conference on information communication and management (icicm). 2016, p. 291–7. http://dx.doi.org/10.1109/INFOCOMAN.2016.7784260.

[42] Williams R, Rojas E, Peek N, Johnson OA. Process mining in primary care: A literature review. Stud Health Technol Inform 2018;247:376–80.

[43] Bogarín A, Cerezo R, Romero C. A survey on educational process mining. Wiley Interdiscip Rev: Data Min Knowl Discov 2018;8(1):e1230.

[44] Yang W, Su Q. Process mining for clinical pathway: Literature review and future directions. In: 2014 11th international conference on service systems and service management (icsssm). 2014, p. 1–5. http://dx.doi.org/10.1109/ICSSSM.2014.6943412.

[45] Rojas E, Munoz-Gama J, Sepulveda M, Capurro D. Process mining in healthcare: A literature review. J Biomed Inform 2016;61:224–36. http://dx.doi.org/10.1016/j.jbi.2016.04.007.

[46] Leitner M, Rinderle-Ma S. A systematic review on security in process-aware information systems - constitution, challenges, and future directions. Inf Softw Technol 2014;56(3):273–93. http://dx.doi.org/10.1016/j.infsof.2013.12.004.

[47] Kelemen R. Systematic review on process mining and security. In: Central and eastern european e| dem and e| gov days 2017. 2017.

[48] Švábenský V, Vykopal J, Čeleda P. What are cybersecurity education papers about? A systematic literature review of SIGCSE and ITiCSE conferences. In: Proceedings of the 51st acm technical symposium on computer science education. New York, NY, USA: Association for Computing Machinery; 2020, p. 2–8.

[49] Husák M, Komárková J, Bou-Harb E, Čeleda P. Survey of attack projection, prediction, and forecasting in cyber security. IEEE Commun Surv Tutor 2019;21(1):640–60. http://dx.doi.org/10.1109/COMST.2018.2871866.

[50] Bakhshi Z, Rodriguez-Navas G, Hansson H. Dependable fog computing: A systematic literature review. In: 2019 45th euromicro conference on software engineering and advanced applications (seaa). 2019, p. 395–403. http://dx.doi.org/10.1109/SEAA.2019.00066.

[51] Buhnova B, Chren S, Fabriková L. Failure data collection for reliability prediction models: A survey. In: Proceedings of the 10th international acm sigsoft conference on quality of software architectures. Qosa '14, New York, NY, USA: Association for Computing Machinery; 2014, p. 83–92. http://dx.doi.org/10.1145/2602576.2602586.

[52] Suryotrisongko H, Musashi Y. Review of cybersecurity research topics, taxonomy and challenges: Interdisciplinary perspective. In: 2019 ieee 12th conference on service-oriented computing and applications (soca). 2019, p. 162–7. http://dx.doi.org/10.1109/SOCA.2019.00031.

[53] Kitchenham B, Charters S. Guidelines for performing systematic literature reviews in software engineering. 2007.

[54] Bernardi S, Trillo-Lado R, Merseguer J. Detection of integrity attacks to smart grids using process mining and time-evolving graphs. In: 2018 14th european dependable computing conference (edcc). 2018, p. 136–9. http://dx.doi.org/10.1109/EDCC.2018.00032.

[55] Myers D, Suriadi S, Radke K, Foo E. Anomaly detection for industrial control systems using process mining. Comput Secur 2018;78:103–25. http://dx.doi.org/10.1016/j.cose.2018.06.002.

[56] Bernardi ML, Cimitile M, Distante D, Martinelli F, Mercaldo F. Dynamic malware detection and phylogeny analysis using process mining. Int J Inf Secur 2019;18(3):257–84. http://dx.doi.org/10.1007/s10207-018-0415-3.

[57] Cimino MG, De Francesco N, Mercaldo F, Santone A, Vaglini G. Model checking for malicious family detection and phylogenetic analysis in mobile environment. Comput Secur 2020;90:101691.

[58] Hluchý L, Habala O. Enhancing mobile device security with process mining. In: 2016 ieee 14th international symposium on intelligent systems and informatics (sisy). 2016, p. 181–4. http://dx.doi.org/10.1109/SISY.2016.7601493.

[59] de Alvarenga SC, Barbon S, Miani RS, Cukier M, Zarpelão BB. Process mining and hierarchical clustering to help intrusion alert visualization. Comput Secur 2018;73:474–91. http://dx.doi.org/10.1016/j.cose.2017.11.021.

[60] Coltellese S, Maggi FM, Marrella A, Massarelli L, Querzoni L. Triage of iot attacks through process mining. In: Otm confederated international conferences" on the move to meaningful internet systems". Springer; 2019, p. 326–44.

[61] Compagna L, dos Santos DR, Ponta SE, Ranise S. Aegis: Automatic enforcement of security policies in workflow-driven web applications. In: Proceedings of the seventh acm on conference on data and application security and privacy. Codaspy '17, New York, NY, USA: ACM; 2017, p. 321–8. http://dx.doi.org/10.1145/3029806.3029813.

[62] Sahlabadi M, Muniyandi R, Shukur Z. Detecting abnormal behavior in social network websites by using a process mining technique. J Comput Sci 2014;10:393–402. http://dx.doi.org/10.3844/jcssp.2014.393.402.

[63] Bernardi S, Alastuey RP, Trillo-Lado R. Using process mining and model-driven engineering to enhance security of web information systems. In: 2017 ieee european symposium on security and privacy workshops (euros pw). 2017, p. 160–6. http://dx.doi.org/10.1109/EuroSPW.2017.66.

[64] Viticchié A, Regano L, Basile C, Torchiano M, Ceccato M, Tonella P. Empirical assessment of the effort needed to attack programs protected with client/server code splitting. Empir Softw Eng 2020;25(1):1–48.

[65] Bahrani A, Bidgly AJ. Ransomware detection using process mining and classification algorithms. In: 2019 16th international isc (iranian society of cryptology) conference on information security and cryptology (iscisc). 2019, p. 73–7.

[66] Macak M, Kruzikova A, Daubner L, Buhnova B. Simulation games platform for unintentional perpetrator attack vector identification. In: Proceedings of the ieee/acm 42nd international conference on software engineering workshops. Icsew'20, New York, NY, USA: Association for Computing Machinery; 2020, p. 222—229. http://dx.doi.org/10.1145/3387940.3391475.

[67] Macak M, Vanat I, Merjavy M, Jevocin T, Buhnova B. Towards process mining utilization in insider threat detection from audit logs. In: 2020 seventh international conference on social networks analysis, management and security (snams). 2020, p. 1–6. http://dx.doi.org/10.1109/SNAMS52053.2020.9336573.

[68] Li C, Ge J, Li Z, Huang L, Yang H, Luo B. Monitoring interactions across multi business processes with token carried data. IEEE Trans Serv Comput 2018;1. http://dx.doi.org/10.1109/TSC.2016.2645690.

[69] Talamo M, Povilionis A, Arcieri F, Schunck CH. Providing online operational support for distributed, security sensitive electronic business processes. In: 2015 international carnahan conference on security technology (iccst). 2015, p. 49–54. http://dx.doi.org/10.1109/CCST.2015.7389656.

[70] Salnitri M, Alizadeh M, Giovanella D, Zannone N, Giorgini P. From security-by-design to the identification of security-critical deviations in process executions. In: International conference on advanced information systems engineering. Springer; 2018, p. 218–34.

[71] Fazzinga B, Folino F, Furfaro F, Pontieri L. An ensemble-based approach to the security-oriented classification of low-level log traces. Expert Syst Appl 2020;153:113386. http://dx.doi.org/10.1016/j.eswa.2020.113386.

[72] Böhmer K, Rinderle-Ma S. Multi-perspective anomaly detection in business process execution events. In: Otm confederated international conferences" on the move to meaningful internet systems". Springer; 2016, p. 80–98.

[73] Huda S, Ahmad T, Sarno R, Santoso HA. Identification of process-based fraud patterns in credit application. In: 2014 2nd international conference on information and communication technology (icoict). 2014, p. 84–9. http://dx.doi.org/10.1109/ICoICT.2014.6914045.

[74] Baader G, Krcmar H. Reducing false positives in fraud detection: Combining the red flag approach with process mining. Int J Account Inf Syst 2018;31:1–16. http://dx.doi.org/10.1016/j.accinf.2018.03.004.

[75] Stahl C, van der Aalst WM. Behavioral service substitution. In: Web services foundations. Springer; 2014, p. 215–44.

[76] Lübke D. Extracting and conserving production data as test cases in executable business process architectures. Procedia Comput Sci 2017;121:1006–13. http://dx.doi.org/10.1016/j.procs.2017.11.130, cENTERIS 2017 - International Conference on EN- TERprise Information Systems / ProjMAN 2017 - International Confer- ence on Project MANagement / HCist 2017 - International Conference on Health and Social Care Information Systems and Technologies, CEN- TERIS/ProjMAN/HCist 2017.

[77] Rubin VA, Mitsyuk AA, Lomazova IA, van der Aalst WMP. Process mining can be applied to software too!. In: Proceedings of the 8th acm/ieee international symposium on empirical software engineering and measurement. Esem '14, New York, NY, USA: Association for Computing Machinery; 2014, http://dx.doi.org/10.1145/2652524.2652583.

[78] Xu X, Zhu L, Weber I, Bass L, Sun D. Pod-diagnosis: Error diagnosis of sporadic operations on cloud applications. In: 2014 44th annual ieee/ifip international conference on dependable systems and networks. 2014, p. 252–63. http://dx.doi.org/10.1109/DSN.2014.94.

[79] Gupta M. Improving software maintenance using process mining and predictive analytics. In: 2017 ieee international conference on software maintenance and evolution (icsme). 2017, p. 681–6. http://dx.doi.org/10.1109/ICSME.2017.39.

[80] Pecchia A, Weber I, Cinque M, Ma Y. Discovering process models for the analysis of application failures under uncertainty of event logs. Knowl-Based Syst 2020;189:105054. http://dx.doi.org/10.1016/j.knosys.2019.105054.

[81] van der Werf JME, Verbeek H. Online compliance monitoring of service landscapes. In: International conference on business process management. Springer; 2014, p. 89–95.

[82] Zahoransky RM, Koslowski T, Accorsi R. Toward resilience assessment in business process architectures. In: International conference on computer safety, reliability, and security. Springer; 2014, p. 360–70.

[83] Corradini F, Marcantoni F, Morichetta A, Polini A, Re B, Sampaolo M. Enabling auditing of smart contracts through process mining. In: From software engineering to formal methods and tools, and back. Springer; 2019, p. 467–80.

[84] Choi S-S, Cha S-H, Tappert CC. A survey of binary similarity and distance measures. J Syst Cybern Inform 2010;8(1):43–8.

[85] Myers D, Radke K, Suriadi S, Foo E. Process discovery for industrial control system cyber attack detection. In: De Capitani di Vimercati S, Martinelli F, editors. Ict systems security and privacy protection. Cham: Springer International Publishing; 2017, p. 61–75.

[86] Maggi FM, Bose RPJC, van der Aalst WMP. A knowledge-based integrated approach for discovering and repairing declare maps. In: Advanced information systems engineering - 25th international conference, caise 2013, valencia, spain, june 17-21, 2013. proceedings. 2013, p. 433–48.

[87] Verbeek H, Buijs J, Van Dongen B, van der Aalst WM. Prom 6: The process mining toolkit. In *Proc. of bpm demonstration track, Vol. 615*, 2010, pp. 34–39.

[88] Günther CW, Rozinat A. Disco: Discover your processes. BPM (Demos) 2012;940:40–4.

[89] Conforti R, La Rosa M, ter Hofstede AH. Filtering out infrequent behavior from business process event logs. IEEE Trans Knowl Data Eng 2016;29(2):300–14.

[90] Fani Sani M, van Zelst SJ, van der Aalst WM. Improving process discovery results by filtering outliers using conditional behavioural probabilities. In: International conference on business process management. Springer; 2017, p. 216–29.

[91] Salnitri M, Paja E, Poggianella M, Giorgini P. Sts-tool 3.0: Maintaining security in socio-technical systems. In: Caise forum. 2015, p. 205–12.

[92] Fazzinga B, Flesca S, Furfaro F, Pontieri L. Online and offline classification of traces of event logs on the basis of security risks. J Intell Inf Syst 2018;50(1):195–230. http://dx.doi.org/10.1007/s10844-017-0450-y.

[93] Fazzinga B, Folino F, Furfaro F, Pontieri L. Combining model-and example-driven classification to detect security breaches in activity-unaware logs. In: Otm confederated international conferences" on the move to meaningful internet systems". Springer; 2018, p. 173–90.

[94] Aalst Wvd. Big software on the run: In vivo software analytics based on process mining (keynote). In: Proceedings of the 2015 international conference on software and system process. Icssp 2015, New York, NY, USA: Association for Computing Machinery; 2015, p. 1–5. http://dx.doi.org/10.1145/2785592.2785593.

[95] Park J, Jung J-Y, Heo G, Kim Y, Kim J, Cho J. Application of a process mining technique to identifying information navigation characteristics of human operators working in a digital main control room - feasibility study. Reliab Eng Syst Saf 2018;175:38–50. http://dx.doi.org/10.1016/j.ress.2018.03.003.

[96] Leppäkoski A, Hämäläinen TD. Promote: A process mining tool for embedded system development. In: International conference on product-focused software process improvement. Springer; 2016, p. 529–38.

[97] Sfyrla V, Carmona J, Henck P. Process-oriented analysis for medical devices. In: Turau V, Kwiatkowska M, Mangharam R, Weyer C, editors. 5th workshop on medical cyber-physical systems. OpenAccess series in informatics (oasics), Vol. 36, Dagstuhl, Germany: Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik; 2014, p. 143–6. http://dx.doi.org/10.4230/OASIcs.MCPS.2014.143, URL http://drops.dagstuhl.de/opus/volltexte/2014/4533.

[98] Fernandez-Llatas C, Ibanez-Sanchez G, Celda A, Mandingorra J, Aparici-Tortajada L, Martinez-Millana A, et al. Analyzing medical emergency processes with process mining: The stroke case. In: Business process management workshops. Cham: Springer International Publishing; 2019, p. 214–25.

[99] Haouari G, Ghannouchi SA. Quality assessment of an emergency care process model based on static and dynamic metrics. Procedia Comput Sci 2017;121:843–51. http://dx.doi.org/10.1016/j.procs.2017.11.109.

[100] Berti A, van Zelst SJ, van der Aalst W. Process mining for python (pm4py): Bridging the gap between process- and data science. 2019, arXiv:1905.06169.

[101] van Zelst SJ, van Dongen BF, van der Aalst WM. Event stream-based process discovery using abstract representations. Knowl Inf Syst 2018;54(2):407–35.

[102] Burattin A, van Zelst SJ, Armas-Cervantes A, van Dongen BF, Carmona J. Online conformance checking using behavioural patterns. In: Weske M, Montali M, Weber I, vom Brocke J, editors. Business process management. Cham: Springer International Publishing; 2018, p. 250–67.

[103] Zhou X, Jin Y, Zhang H, Li S, Huang X. A map of threats to validity of systematic literature reviews in software engineering. In: 2016 23rd asia-pacific software engineering conference (apsec). 2016, p. 153–60.