IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 14 August 2022, accepted 14 September 2022, date of publication 21 September 2022,
date of current version 29 September 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3208231

RESEARCH ARTICLE

Complex Process Modeling in Process Mining: A Systematic Review

MOHAMMAD IMRAN[1,2], (Member, IEEE), MAIZATUL AKMAR ISMAIL[1], (Member, IEEE), SURAYA HAMID[1], (Member, IEEE), AND MOHAMMAD HAIRUL NIZAM MD NASIR[3], (Member, IEEE)

[1]Department of Information Systems, Faculty of Computer Science and Information Technology, University of Malaya, Kuala Lumpur 50603, Malaysia
[2]Department of Information Technology, Faculty of Information and Communication Technology, Balochistan University of Information Technology, Engineering and Management Sciences, Quetta 87300, Pakistan
[3]Department of Software Engineering, Faculty of Computer Science and Information Technology, University of Malaya, Kuala Lumpur 50603, Malaysia

Corresponding authors: Maizatul Akmar Ismail (maizatul@um.edu.my) and Mohammad Imran (mimranit@outlook.com)

ABSTRACT Process mining techniques are used to extract knowledge about the efficiency and compliance of an organization's business processes through process models. Real-life processes are unstructured, and applying process mining to discover such processes often results in complex process models that do not provide actionable insights. Several solutions have been presented to overcome this problem. However, the process mining domain lacks an explicit definition of complexity and its measurement. This vagueness results in ad-hoc solutions that vary according to the approach, modelling construct, and process properties. Additionally, the strength and limitations of the proposed solutions have not been adequately highlighted. Therefore, we conducted a systematic literature review on complexity in process mining over six popular scholarly literature indexing databases. Based on the review results, an explicit definition of complexity, the main contributing factors and their impact on process mining results were identified. We discovered various process complexity matrices and their application context. The analysis of studies led to the development of a taxonomy consisting of four different approaches for addressing the complexity problem, along with their strengths and limitations. Finally, the open research challenges and potential for future research are discussed.

INDEX TERMS Complexity, complex process models, complex process mining, process management, process mining, systematic literature review.

I. INTRODUCTION

The current age of technology has significantly changed how an organization manages its business operations. Organizations have shifted from manual processing to automated and technological methods of business operations. Information systems are used almost everywhere, from banks to hospitals. With the increased usage of technology for information management, there has been an increase in data generation. This outburst of data introduces difficulty for organizations to extract valuable insights from these systems. Regardless of the statistical analysis techniques to assess business operations, it is also crucial for an organization to know how efficiently their business operates, where and why bottlenecks exist, and how they can be removed. Although Data mining techniques can uncover certain patterns in the data of business operations, no temporal relationship exists between such data. An end-to-end multi-perspective process execution insights are not possible using data mining techniques [1]. To make informed decisions, even a data scientist finds it crucial to analyze the relationship between data and business operational processes, which is not possible without a holistic understanding of the underlying process [2].

Process mining (PM) is an umbrella term for combining the data mining and business process management approaches that analyze event log data using advanced algorithms, machine learning, and statistical methods to analyze and improve business processes. PM techniques extract the

The associate editor coordinating the review of this manuscript and approving it for publication was Fu Lee Wang.

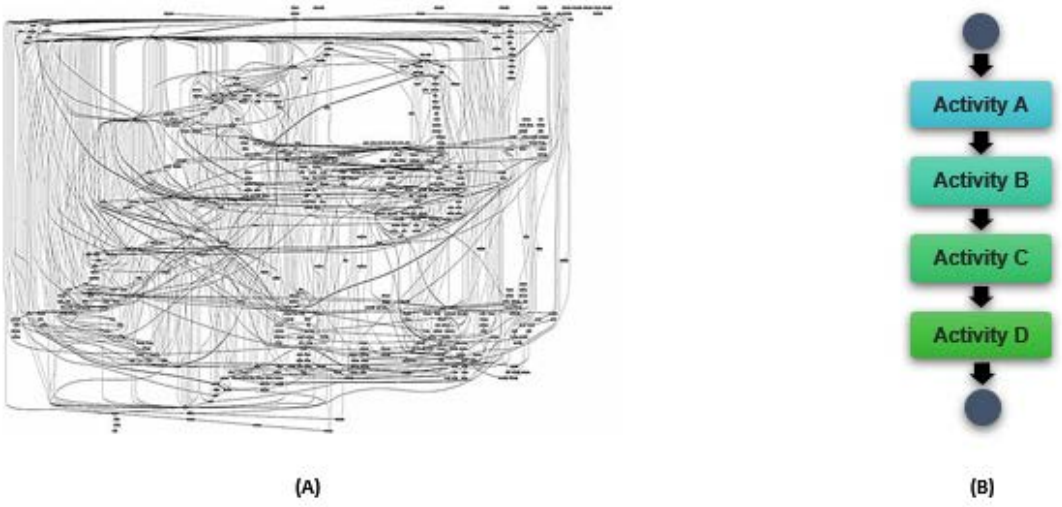

FIGURE 1. (A) Spaghetti model originated from a complex and unstructured process (B) structured model originated from a strictly defined process.

knowledge related to processes by using logs generated by systems, also called event logs or process execution logs. Event logs can be considered as the logbooks recording each process execution step. Such process-related knowledge is represented in a graphical form known as the process models [3], also referred to as mined models that describe the process execution behavior [4]. One of three perspectives of process mining, Process Discovery is used to discover a process model from event logs and is usually the foundation of the subsequent analysis [5]. The discovered process model is used to analyze inefficiencies in business processes and obtain more profound knowledge of their root causes and influence on key performance metrics. Since processes are first modeled using process discovery techniques, this implies that if process discovery was not carried out correctly, the process discovery results (control flow, time performance analysis) and subsequent analysis will lead to misleading results.

Over time, several process discovery algorithms were developed. Alpha miner [2], Heuristic miner [6], Inductive miner [7], and the Fuzzy miner [8] are the most prevalent process discovery algorithms [5]. However, the resulting models are complex and challenging to interpret when discovering complex or unstructured process execution behaviors. Such unpredictive behavior of process models resulted from the assumption that experiments conducted in a controlled environment will also work for real-world data. However, real-life process logs contained traces of flexible behavior, which allowed unpredicted results [9], [10]. Consider an example of an organization that asks its employees to achieve a particular target containing several activities with no restriction on the order of activities to follow. In such a case, the employees can follow any combination of activity sequences. The execution of such a process will result in various traces where one trace may contain five activities whereas in the other case, it may contain twenty-five. Such a variable process is called a complex or unstructured process. The model generated by process mining using logs of such an environment result in a complex or spaghetti-like process model [8], [11]. Examples of both the complex and structured process models are shown in Fig. 1. The process model on the right i.e., structured model is easy to understand and clearly conveys the flow of the process. In contrast, the model on the left is overly complex and does not provide any insight into the process execution flow.

According to Mendling *et al.* [12] and Reijers and Mendling [13], as cited in Li *et al.* [14], a human analyst's ability to understand a process model is known to be influenced by the complexity and density of a process model. So, to understand and improve processes, process models must not be overly complex and should be easy to understand [15].

Several researchers have attempted to resolve this problem. However, a general understanding of the complexity problem, what causes this problem, and what approaches can be used to mitigate this problem remains missing. Reviewing the literature on the complexity in process mining will help to better understand the primary factors that contribute to complexity and possible resolution strategies. A broader analysis of the strengths and limitations of the complexity reduction approaches is also essential to understand the suitability of the approach. Furthermore, the identification of research gaps will help in identifying the untapped research areas. We conducted a systematic literature review on complexity in the process mining domain to answer the previously mentioned questions. Six popular scholarly indexing databases were systematically searched, specifically focusing on published papers between 2012 and 2022. In addition to the above

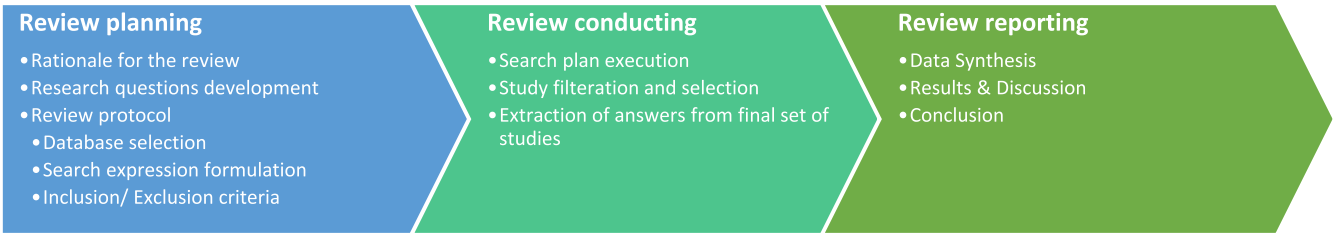

FIGURE 2. Review process with subactivities.

questions, a taxonomy was formulated detailing different approaches and sub-approaches to deal with complexity. Finally, the prospects of future research in this dimension were identified. The rest of this paper is structured as follows:

Section 2 presents the methodology for planning, conducting, and reporting this review. Section 3 shows results and related discussion. Section 4 puts forward the limitations of this review, and finally, in section 5, we conclude our findings.

II. METHODOLOGY

The systematic review guidelines by Kitchenham *et al.* [16] were followed to perform this review. Three steps according to the guidelines were used, review planning, conducting, and reporting. The focus of the planning stage was to develop a review protocol and formulate the research questions, the search strategy, the selection of databases to search, and the inclusion and exclusion criteria. The search plan is executed in the second phase, the study screening process is carried out, and relevant answers from selected studies are extracted. In reporting stage, the review results are disseminated. The review process is presented in Fig. 2.

A. REVIEW PLANNING

This section contains further subsections, the review rationale, research questions, and research protocol as detailed in subsequent sections.

1) THE RATIONALE FOR THE REVIEW

A considerable number of research works exist to deal with the problem of complexity in process mining, proposing diverse approaches to deal with complexity. Studies of complexity reduction and similar techniques were carried out by La Rosa *et al.* [17] and Schonenberg *et al.* [18]. However, these reviews are obsolete as they were published early when process mining techniques were still flourishing. Secondly, their focus was complexity reduction techniques focused on block-structured process models and specific modeling languages such as YAWL modeling notation. Methods have evolved since then, and the focus has shifted from only a single modeling notation to other ways of presenting a process model. Houy *et al.* [19] conducted a literature review on complexity challenges faced by Business Process Management (BPM) community. Their focus remained on complexity introduced in the general BPM domain rather than process mining. D'Castro *et al.* [20] investigated whether a low-structured process can be modeled using process discovery techniques. However, their research was one of the experimental studies of applying process maps, whose results are already known earlier as conducted by Günther and Van Der Aalst [8]. A systematic review was conducted by Duan and Wei [21], focusing only on the complexity caused by duplicate tasks in process mining. Van Zelst *et al.* [22] performed a literature review on abstractions in process mining and presented a taxonomy of works on event abstraction techniques. Nevertheless, their assessment was explicitly focused only on event abstraction.

Although a fair bit of literature exists on the topic, there is no systematic review of the diverse approaches dealing with the problem of complexity in the process mining domain. Almost every proposed approach to deal with complexity in process mining holds a different view on this problem, and there is a lack of a unified view of the topic. Therefore, this research aims to fill this void by systematically reviewing the available literature and presenting the strengths and limitations of existing approaches and opportunities for future works, along with the taxonomy of different approaches used to resolve this problem.

For this purpose, we formulated the following research questions and sub-questions as presented in following section.

2) RESEARCH QUESTIONS

RQ 1. What is a complex process in process mining throughout the process mining literature?
  1. Why does a process become complex?
  2. What is the impact or the consequence of the complexity?

RQ 2. How do researchers measure process complexity?

RQ 3. Which techniques are used to deal with process model complexity?

RQ 4. What are the strengths and limitations of the techniques that deal with the complexity?

RQ 5. What are the open research challenges and avenues of future research in this direction?

3) REVIEW PROTOCOL

The review protocol for this review consists of search string formulation, selecting suitable databases to search, and

formulating the inclusion and exclusion criteria. The details of each section are presented in the following subsections.

a: SEARCH STRINGS

After performing mockup searches on the selected databases, the search expression returning the relevant results was identified. The Boolean operator ''AND'' helped narrow the search space to the process mining domain, whereas using the ''OR'' operator included synonymous words. Finally, the database-specific version of the following search expression was used for searching.

("process mining") AND ("complex process" OR "complex processe" OR "unstructured process" OR "unstructured processes" OR "flexible process" OR "flexible processes" OR "spaghetti model" OR "spaghetti process model" OR "complex model" OR "complex process model")

b: DATABASES TO SEARCH

Six online scholarly databases were selected for literature search, i.e., the Web of Science, Scopus, Science Direct, IEEE Xplore, Springer Link, and Google Scholar.

Web of Science (WoS) was selected for its quality journal indexing criteria. The Scopus was also added because some WoS unindexed papers are indexed in Scopus as it indexes abstracts and references from thousands of other publishers, including Elsevier. Since Scopus does not index full-texts, we added Science Direct as it indexes full-text articles from journals and books, mainly published by Elsevier and a few other sources. To also include conference proceedings, we added IEEE Xplore and SpringerLink. SpringerLink was selected because of its popularity in the computer science domain and conference proceedings indexing in well-known springer lecture notes series such as LNCS, LNBI, LNBIP, and others. Due to quality factors, WoS and Scopus leave out publications from less popular sources. To ensure that no research work is left out, Google Scholar (GS) was also included. However, it is worth mentioning that because of indexing the grey literature, researchers do not recommend GS as a primary source of systematic literature searching [23]. We take care of such quality issues in our inclusion and exclusion criteria. Since GS does not provide a search results extraction feature [23], we used ''Publish or Perish'' (PoP) by Harzing [24], a search results retrieval tool that made GS search results extraction possible.

c: INCLUSION AND EXCLUSION CRITERIA

The inclusion criteria (IC) refer to selecting papers that fulfill requirements, whereas the Exclusion criteria (EC) are constraints to remove articles that do not meet specific requirements. The IC set for this review comprises the articles published between 2012 and 2022 and available online. The selection of the last ten years' literature was motivated by an urge to review all the past attempts to deal with the problem at hand and the maturity of the process mining domain. The IC also enforces the selection of only the conference proceedings and journal articles. Furthermore, some significantly important papers were added by snowballing that remain unaffected by the duration bound.

TABLE 1. Inclusion criteria.

Inclusion Criteria	Description
IC 1	Electronically available and found by the search string within the specified period
IC 2	Conference proceedings and Journal papers
IC 3	Online publication during 2012 – 2022
IC 4	Referenced papers not indexed in selected digital libraries (snowballing)
IC 5	Papers published before 2021 should have at least one citation, whereas 2021-2022 papers are exempted from this requirement to prevent selection bias towards newly published articles that are yet uncited

TABLE 2. Exclusion criteria.

Exclusion Criteria	Description
EC 1	Papers in languages other than English
EC 2	Duplicate papers
EC 3	Workshop papers or papers merely published in the institutional repository without any publication in conference or Journal
EC 4	Experimental papers on mere usage of process mining
EC 5	The study does not contribute to process complexity reduction or other complexity, such as enterprise complexity
EC 6	Unavailable Papers
EC 7	Process complexity merely mentioned, e.g., in the author intro, introduction, and other sections
EC 8	Off-topic, dealing with process complexity using techniques other than process mining domain

Table 1 shows the inclusion criteria, whereas the exclusion criteria are presented in Table 2.

B. REVIEW CONDUCTING

In this stage, the review plan was executed. The search results were extracted in respective formats (BIB/ CSV/ CIW). For Google scholar specifically, the ''Publish or Perish'' [24] was used to search and extract search results. All the search results were imported into the Zotero reference manager. The inter and intra-database duplications were resolved based on the

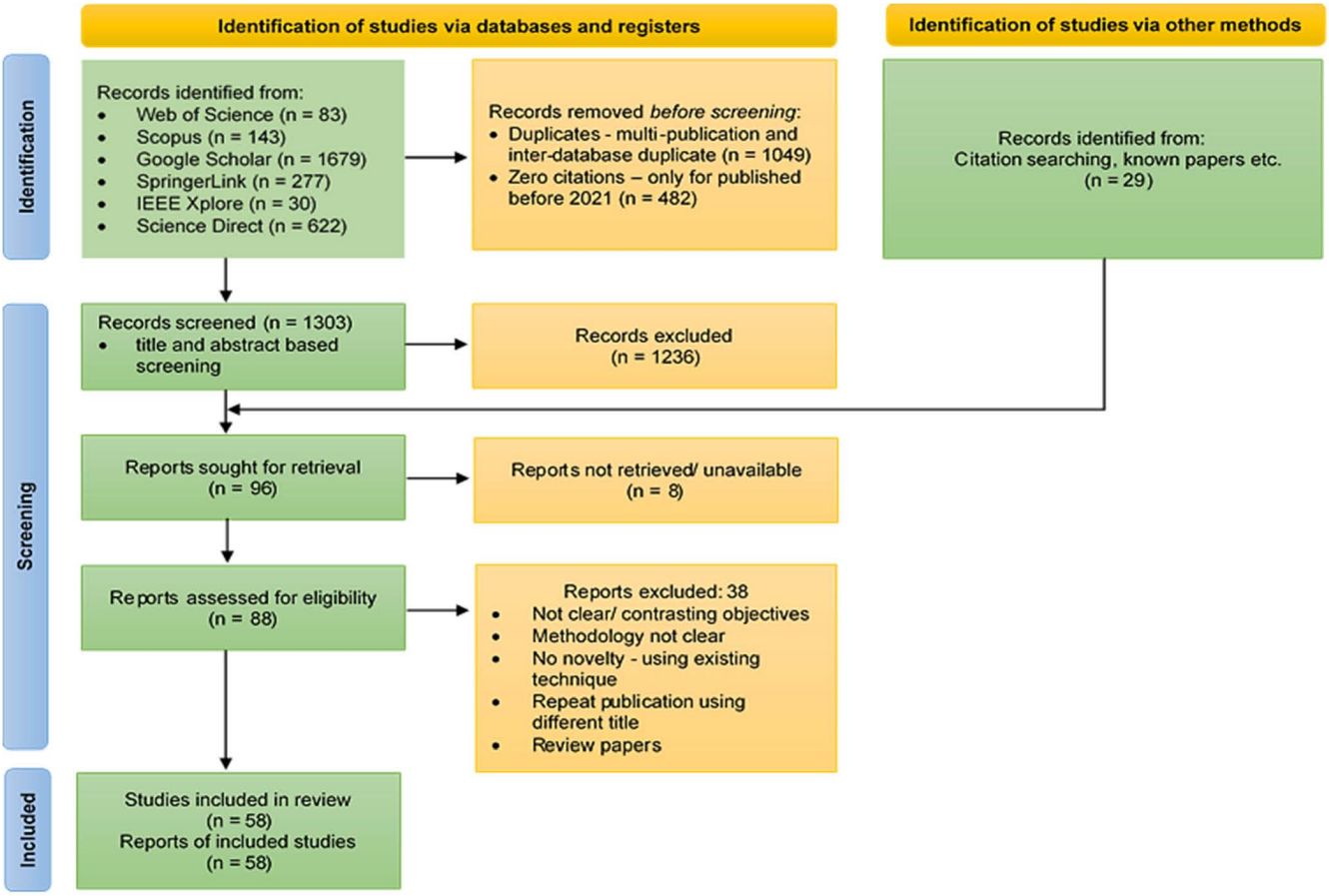

FIGURE 3. PRISMA Flow Diagram. Study inclusion and exclusion Flow chart.

article titles, and the remaining results were exported into a Microsoft Excel sheet. The title and abstract-based screening were performed here based on IC & EC. Finally, the full texts of selected papers were downloaded and exported into the Mendeley reference manager. We used an Excel sheet for quantitative analysis of extracted answers. Fig. 3 shows each step of the screening process reported in the PRISMA flow diagram as per the standard study selection and reporting method for systematic reviews [25], [26].

Initially, the search resulted in 2834 results. Such a high number of initially retrieved results is attributed to multiple indexes of the same articles in different databases. The inter-database and intra-database deduplication helped in resolving the redundancy. To maintain quality, we only selected papers having at least one citation. However, this resulted in a bias towards newly published studies, so a minimum one citation rule was relaxed for recently published studies, i.e., studies published in 2021 and 2022. Afterward, the title and abstract screening were performed on 1303 remaining articles using IC & EC, which resulted in the selection of 96 articles and some already known papers. Upon full-text review, 38 articles were deemed ineligible because of repeat publication, mere review papers, contrasting objectives and methodology, or non-novel technique. Finally, 58 papers were deemed eligible.

1) DATA COLLECTION AND ANALYSIS

A form was designed in a Microsoft Excel sheet to collect answers corresponding to the research questions. While performing full-text screening, such papers were deemed ineligible whose objectives were unclear, remained ambiguous about methodology, and were weak on novelty perspective. Furthermore, to maintain the quality, the articles with replicated publications were excluded, in addition to those with contrasting claims such as title and abstract indicated novelty but the paper's body suggested otherwise.

C. REPORTING THE FINDINGS

We report the results of our findings based on the questions this review intends to answer. In the results and discussion section, the answers to the questions are first presented in frequency-based quantitative analysis format. A taxonomy of approaches/ techniques that deal with process model complexity in process mining is formulated. Moreover, we also elaborate on the interpretation of the results based on the review.

III. RESULTS

This section presents the findings in answers to the format of the questions. The results are presented visually in quantitative format along with the corresponding interpretations.

A. RQ 1: WHAT IS A COMPLEX PROCESS IN PROCESS MINING THROUGHOUT THE PROCESS MINING LITERATURE?

To understand the problem of dealing with the complex processes in process mining, firstly, it is necessary to understand what a complex process is? The factors leading to Process complexity and impact of the complex process on process mining.

As a result of our analysis, complexity in a process is termed in many different ways, such as unstructured, flexible, and fine-grained processes being the primary terms. These terms have been used synonymously in the literature. The abstracted version is shown in Fig. 4. The complexity of a process is defined in the context of specific properties of the event log and the resulting model extracted from such log; for example, if an event log or model holds such properties, it can safely be termed a complex process.

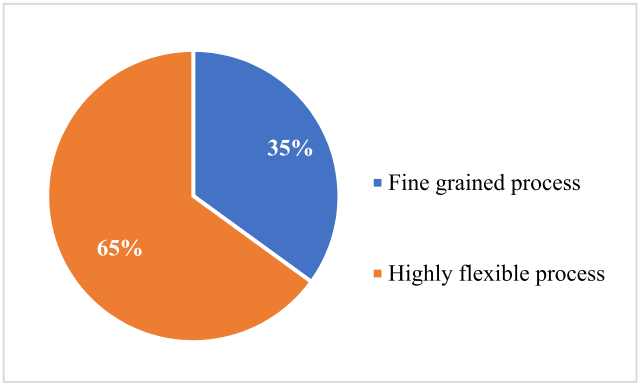

FIGURE 4. Terms for complex process.

Concerning the properties of the event log used for process mining, the following processes can be termed complex processes:

- A highly flexible process, i.e., the process executed in a less restricted environment, resulting in a heterogeneous or high number of variable behaviors in the log. For example, it is defined what tasks must be performed, but there is no restriction imposed on the order of execution of such tasks.
- Fine-grained processes, i.e., processes containing too much detail about process execution, such as precise details about process execution, results in a high number of activities

Concerning the results of applying process mining on the log, the following processes are termed complex processes

- Processes on which applying process mining result in a spaghetti-like process model structure
- The model discovered from the process is not comprehendible, i.e., the inability to understand the model and how the process was executed.

In the light of observing the above two perspectives on considering a process a complex, we deduce the definition of process complexity as:

A process can be termed a complex process if any of the following conditions hold:

- Processes executed in a less restricted environment
- A process whose event log contains a fine-grained level of detail about process execution
- A process in which applying process mining results in spaghetti-like visualization, complicating the understandability of the process execution.

From the above definition, we can safely say that the complexity in process mining is the level of difficulty in bringing simplicity to processes. Please remember that the terms "complex," "flexible," and "unstructured" processes are used synonymously throughout the process mining literature. So, we also use these terms interchangeably throughout this paper.

1) WHAT ARE THE FACTORS THAT INTRODUCE PROCESS COMPLEXITY?

Following three significant factors were found to be the reasons for introducing complexity to the process, also shown graphically in Fig. 5.

a: FLEXIBILITY AND VARIATION IN PROCESS EXECUTION BEHAVIOR

Flexibility in process execution remains the top reason for introducing complexity to the process where there is no restriction imposed on the execution behavior of the process. The process execution behaviors will increase with the number of activities. Even if an order-preserving constraint is imposed on a few activities, a fraction of process execution behaviors still increases with the number of activities. Modeling such a dynamic behavior will result in a complex process model.

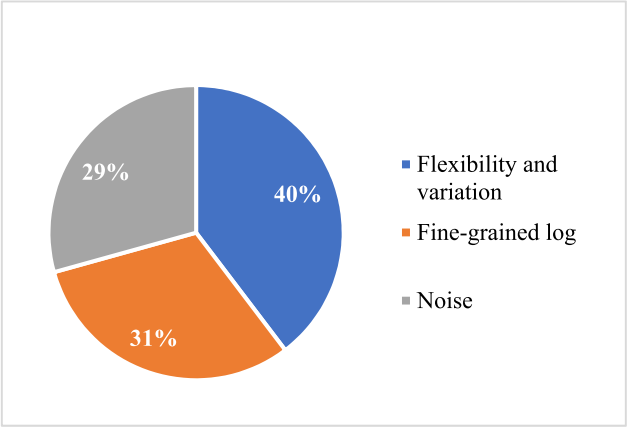

FIGURE 5. Factors responsible for introducing complexity.

b: FINE-GRAINED LEVEL OF DETAIL ABOUT PROCESS EXECUTION

The second common factor that introduces complexity in the process is a fine-grained level of information about process execution. As mentioned previously, such a pattern in the log is the inherent presence of behavior more than the required level. Let us say such a process is mined for an organization that wants to view how the process is executed. The level of process execution-related information will differ based on the hierarchy in the process. If all behavior is included in the process model, this will render the process model unusable, considering that all executive levels do not require the same granularity of information.

c: NOISE IN THE LOG

The Noise in the log was observed as the third most dominant factor contributing to process complexity. In process mining terms, noise is commonly defined as infrequent behavior [4]. Most of the process discovery algorithms model the most frequent behaviors in the process [8], [27], [28]. For example, a path from activity A to B may appear 100 times in a log. At the same time, some paths may infrequently occur, such as activity A to C being observed three times and activity B to A being observed five times (looping pattern). Process discovery from such a log will result in spaghetti-like visualization, which complicates the understandability of the process execution behavior.

According to Conforti *et al.* [29], models discovered using noisy logs tend to be more complex because of the increased number of arcs and nodes resulting from the noise. This behavior is termed noise in the log. It can result from mistakes made during log recordings, such as the incorrect timestamp of process execution, approximate values, labeling, and spelling mistakes. At the same time, few other researchers have defined noise as "chaotic activities," i.e., activities that do not have a specific position and occur randomly [30], [31]. The occurrence of activity at random positions and consequently introducing complexity to the process justifies our definition of complexity as logs originating from flexible and unrestricted environments.

2) WHAT ARE THE CONSEQUENCES OF THE COMPLEXITY WHILE PERFORMING PROCESS MINING?

As per literature, the following impacts can be expected from applying process mining when a process becomes complex due to the factors mentioned earlier. Our quantitative analysis revealed three main categories of impacts on results when performing process mining, as visualized in Fig. 6.

a: COMPLEX VISUALIZATION - SPAGHETTI EFFECT

Firstly, the process model derived from complex process results in a kind of complex visualization known as the Spaghetti process model, and the effect is known as the Spaghetti effect. The name spaghetti refers to the resemblance of the resulting process model to the spaghetti-like structure, which leads to difficult-to-understand process models.

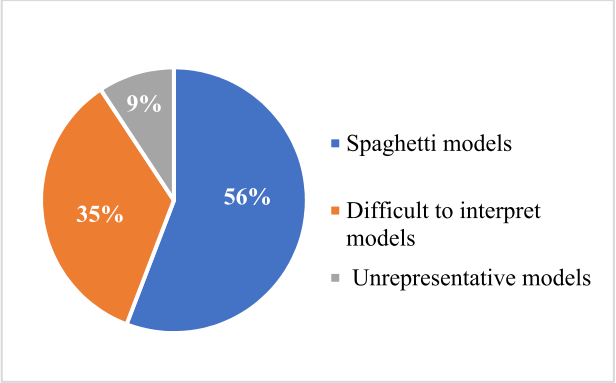

FIGURE 6. Impact of complexity.

b: DIFFICULT TO INTERPRET MODELS

As a consequence of spaghetti effects, the process model becomes almost impossible to comprehend. Since the investigation of process execution behavior is one of the main goals of process mining [2], the complex spaghetti visualization results in a difficult-to-understand process. No actionable knowledge (based on which further actions can be taken) can be acquired from such visualizations [32], [33].

c: INACCURATE MODELS

Another impact of complex process mining is the inability to render an accurate process model. Noise introduces erratic connections in the process model that are never executed in reality, thus resulting in a false reflection of reality [30].

B. RQ 2: HOW DO RESEARCHERS MEASURE PROCESS COMPLEXITY?

Several metrics have been proposed in the literature. This research focuses on metrics relevant to process model quality and complexity measurement. Table 3 presents the different complexity metrics found in the literature.

We categorize process evaluation metrics into two major categories based on their goals:

1. Behavioral complexity
2. Visual/ Structural complexity

1) BEHAVIORAL COMPLEXITY (ALSO BEHAVIORAL CORRECTNESS/SIMILARITY) METRICS

Behavioral complexity (also behavioral correctness/similarity or behavioral appropriateness) metrics are used to measure the presence of correct behavior in mined model with reference to the original model or log. It is a measure to check the correspondence of mined model with reference behavior [34]. The metrics to measure model correctness are visualized in Fig. 7.

TABLE 3. Behavioral complexity and structural complexity metrics.

Category	Metric
Behavioral complexity metrics	Fitness Precision F-Score Generalization Number of subprocesses
Structural complexity metrics	Number of arcs and nodes Just visual analysis Density Control Flow Complexity (CFC) Coefficient of Network Connectivity (CNC) Place/Transition Connection Degree (P/T-CD) Cyclomatic Number (CN) Average Connector Degree (ACD) Average number of activities per subprocess Number of event classes and variants

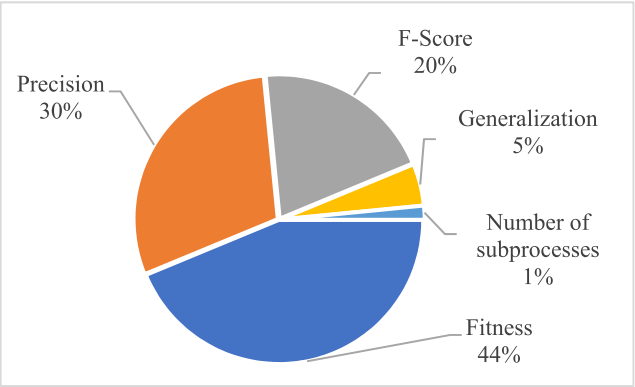

FIGURE 7. Behavioral complexity measures.

We elaborate on different behavioral complexity metrics along with their calculation formulas found in the literature as below:

a: FITNESS, PRECISION, AND GENERALIZATION

The most common metric in this direction is the Fitness measure (also called replay fitness, recall, or behavioral recall). Fitness measures the amount of behavior in a reference model that is also present original model or log [4]. Suppose the log contains a behavior A > B where A is directly followed by B, the fitness measures whether such behavior exists in the mined model. It is measured between the range of 0 and 1. The fitness value closer to 1 indicates a more similarity of the mined model to the reference model.

The second most prevalent metric that is usually measured along with the fitness metric is the Precision metric (also called behavioral precision). Precision measures the model specificity, e.g., How precise the mined model is regarding the reference model [4], [35]. It is also measured between the 0 and 1 range, with a value closer to 1 indicating the more specific behavior in the reference model. According to Conforti *et al.* [29], noise significantly reduces model precision because it establishes erroneous links between model activities. The Generalization is the inverse of the precision metric. It measures how much additional behavior is observed in the mined model, which is not present in the log and vice versa [29]. The simplified version of formulas found in the literature for calculating the fitness and precision measures are presented in (1) and (2), respectively.

$$fitness = \frac{|behaviors\ in\ the\ mined\ model|}{|behaviors\ in\ the\ reference\ modelor\log|} \tag{1}$$

$$precision = \frac{|behaviors\ common\ between\ model\ and\ log|}{|behaviors\ in\ the\ mined\ model|} \tag{2}$$

b: F-MEASURE

F-measure, also called f-score or f1-score, is the harmonic mean of fitness and precision, as shown in (3). Researchers observed that if a particular behavior is excluded from the model, the fitness value decreases, increasing the precision (accuracy) value. Considering the trade-off between fitness and precision metrics, researchers proposed the f-measure as an alternative to balancing the model fitness and precision [36].

$$2 * \frac{fitness * precision}{fintness + precision} \tag{3}$$

If a technique produces multiple models, e.g., clustering or abstraction-based approaches, it is logical to use the average fitness, precision, or f-measure [37], [38].

c: NUMBER OF SUBPROCESSES

The number of subprocesses metric is explicitly related to the abstraction-based approaches where a process is simplified by dividing it into subprocesses and assessing whether the formed subprocess relates to the reference model [39].

2) VISUAL (STRUCTURAL) COMPLEXITY MATRICES

Visual or structural complexity metrics measure the Spaghetti-ness of the process model, which is directly related to comprehension of the process model. It measures the simplicity dimension of the resulting process model, i.e., How easy the mined model is to understand [40]. It is measured based on the size of the process model, such as the number of arcs and (or) nodes, Control Flow Complexity (CFC), Average Connector Degree (ACD), Density, and other such metrics [40] as visualized in Fig. 8.

a: NUMBER OF ARCS AND NODES

In a process model, the activities represent nodes, and the relationship between two activities is portrayed as an arc between them. For example, if activity A is followed by Activity B, the resulting process model will contain an arc between activity A and B. As the number of activities and their relationship increases, the process model becomes complex, resulting in the Spaghetti model leading to a less understandable model. The total number of arcs and nodes is the

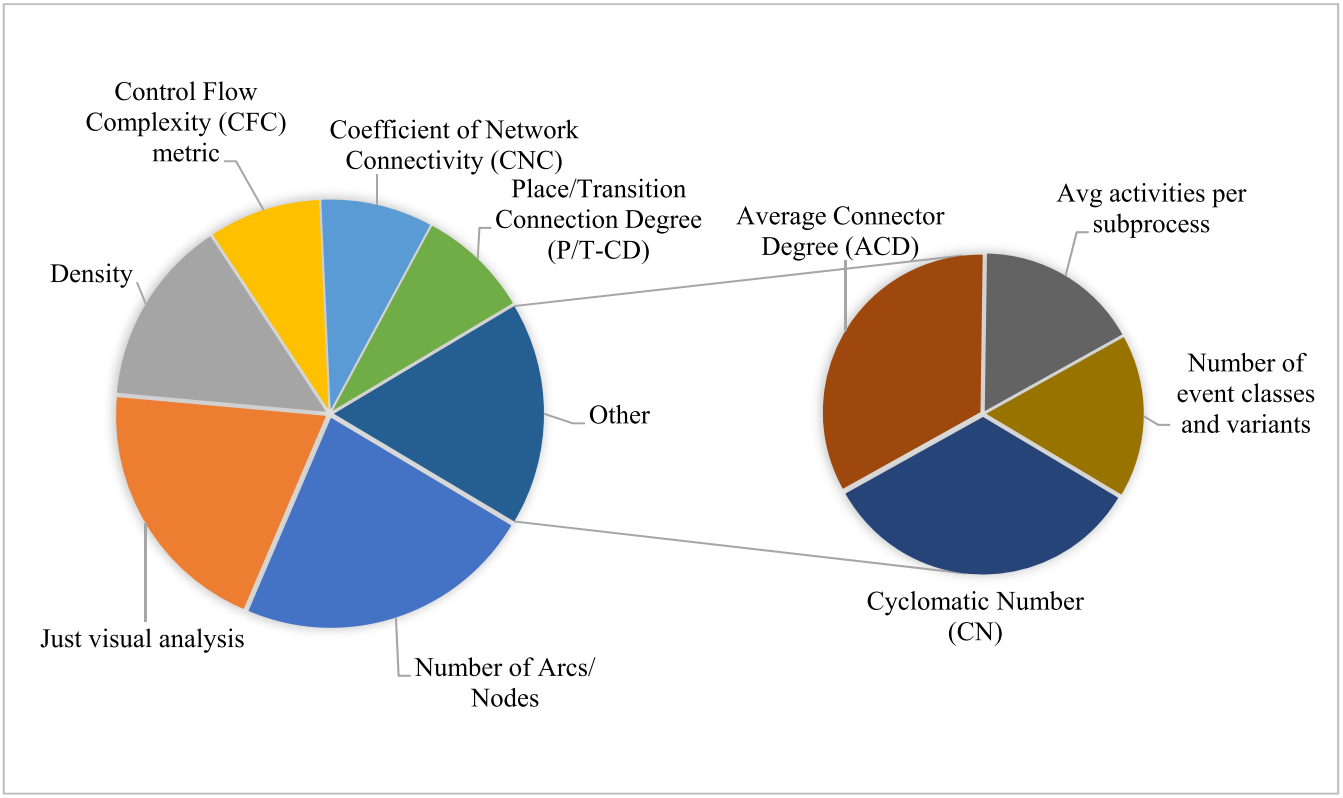

FIGURE 8. Structural complexity metrics.

most prevalent measure to measure the visual complexity of the process model. Since the number of arcs and nodes negatively correlates with the understandability of the process model [41], the more the number of arcs and nodes in the model, the more complex the model is.

b: DENSITY

Another similar measure is density, a ratio measurement among the number of arcs and the highest possible arcs [40], [42], [43], [44]. The higher density represents a more complex process model. The formula for density measurement is reported in (4), where A represents arcs, and N represents nodes

$$Model\ density = \frac{|A|}{|N| * |N-1|} \tag{4}$$

c: THE CONTROL FLOW COMPLEXITY (CFC)

The Control Flow Complexity (CFC) deals with complexity introduced in the model by the presence of gateway/split constructs such as "OR," "XOR," and "AND" [40]. The presence of such constructs in the model results in splits, and an increased number of splits results in a higher number of arcs and more complexity [42]. So, the CFC quantifies the number of arcs going out from each of such constructs [29]. CFC is specifically relevant when the process model representation is a Petri net [29]. The CFC measurement formula is shown in (5).

$$CFC = \sum All\ split\ constucts\ in\ process\ model \tag{5}$$

d: COEFFICIENT OF NETWORK CONNECTIVITY (CNC)

The Coefficient of Network Connectivity (CNC) measures the ratio between the number of arcs and nodes [40]. Parts of the process model containing cycles tend to be more challenging to understand than sequential ones. So, the increase in cycles results in the rise in complexity of the process model. Refer to (6), where |**A**| represents the total number of arcs whereas |**N**| represents the total number of nodes in the model.

$$CNC = \frac{|\boldsymbol{A}|}{|\boldsymbol{N}|} \tag{6}$$

e: PLACE/ TRANSITION CONNECTION DEGREE (P/T-CD)

Place Transition Connection Degree (P/T-CD) is the weighted sum of the average number of arcs per transition and the average number of arcs per place [45]. An increase in arcs connecting places and transitions results in spaghetti-ness and renders an incomprehensible model. The higher value of P/T-CD indicates an increase in the complexity of the process model [46], [47]. In the P/T-CD formula, as shown in (7), |A| represents the total number of arcs in the model, |P| represents the number of places, and |T| represents the

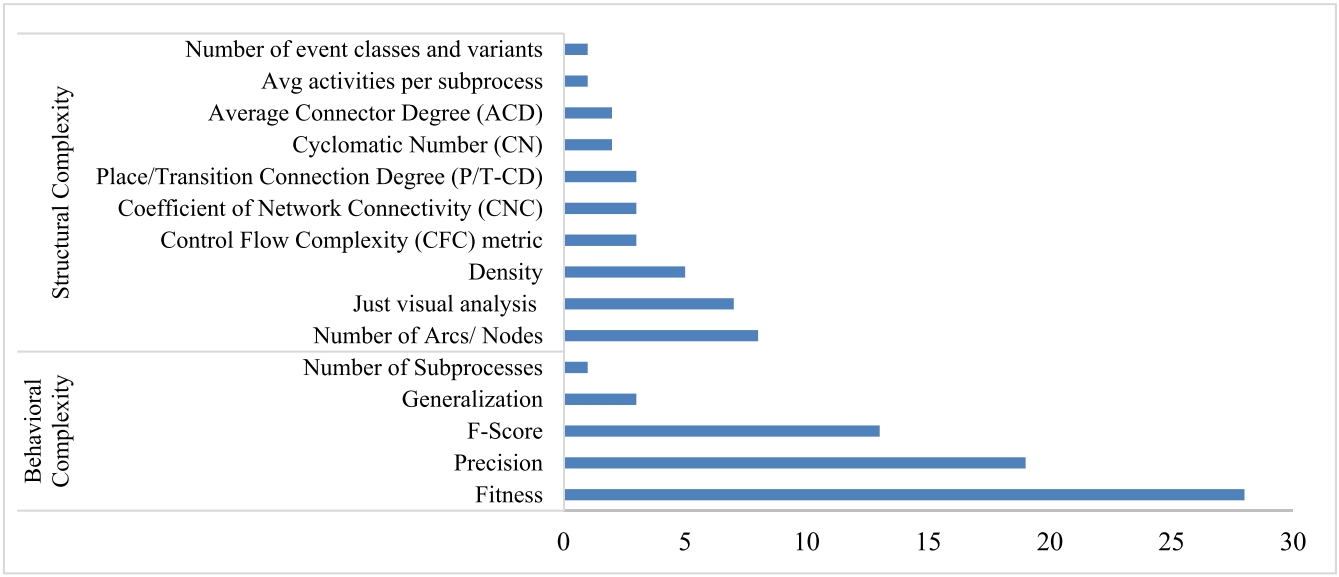

FIGURE 9. Usage of structural complexity vs. behavioral complexity across studies.

number of transitions.

$$P/T - CD = \frac{1}{2}\frac{|A|}{|P|} + \frac{1}{2}\frac{|A|}{|T|} \tag{7}$$

f: CYCLOMATIC NUMBER (CN)

A cyclomatic Number (also called Cyclomatic Complexity) is the number of linearly independent paths in a process model where directions of the arcs are ignored [43], [44]. An increase in cyclomatic number means an increase in branching (splits) of the process model and implies an increase in process model complexity. A process model with a low number of branching will be easier to understand. The cyclomatic number calculation formula is reported in (8).

$$CN = |A| - |N| + 1 \tag{8}$$

g: AVERAGE CONNECTOR DEGREE (ACD)

Average Connector Degree (ACD) measures the average number of connecting nodes with a connector [40], [41]. It represents the average count of incoming and outgoing arcs of places/ transitions. According to Leemans *et al.* [41], a significant negative correlation between both the number of nodes & edges (#(NE)), ACD, and understandability can be observed. So, increased NE and ACD are good indicators of complexity.

h: THE AVERAGE NUMBER OF ACTIVITIES PER SUBPROCESS AND NUMBER OF EVENT CLASSES AND VARIANTS

This metric is specifically relevant in abstraction-based approaches where a model is abstracted to different levels. The average number of activities per subprocess is then calculated to quantify the simplicity introduced by abstraction [39]. The number of event classes and process variants is the count of unique process variants and unique activity classes. According to Baier *et al.* [28], a drop in the activity classes and variants was observed after performing abstraction, resulting in complexity reduction of the process model because of a lower variation.

The frequency of different matrices found for both the structural and behavioral complexity dimensions is presented in Fig. 9. Although most of the studies claim that structural complexity has been taken care of, our analysis, as shown in Fig. 9, reveals that significantly less emphasis has been given to structural complexity metrics. The x-axis shows the usage of each metric in selected studies, whereas the y-axis represents the name of each metric.

Among the found process complexity metrics, the number of nodes and edges was the most straightforward way of measuring process complexity, and it remained the dominant structural complexity metric throughout the literature. Also, it poses the advantage of being feasible for both the Block-structured and graph-based process models. Based on the studies' data, the number of nodes and edges and the Average Connector Degree (ACD) were negatively correlated with understandability [41]. The Cyclomatic complexity metric is another potential measure having its roots in the software quality domain, where it is used to quantify the number of possible code execution paths. It is said that the more control structures in the code, the more branching and complex the code. Similar to complexity in a code caused by control structures, the split constructs cause the complexity and branching in process models. Cyclomatic complexity is equally helpful for process model complexity analysis [44] since the basic idea of complexity and understanding is similar in both cases. It was further found that an increase in places, transitions, and precisely the number of splits and joins affect the comprehensibility of the process model [47].

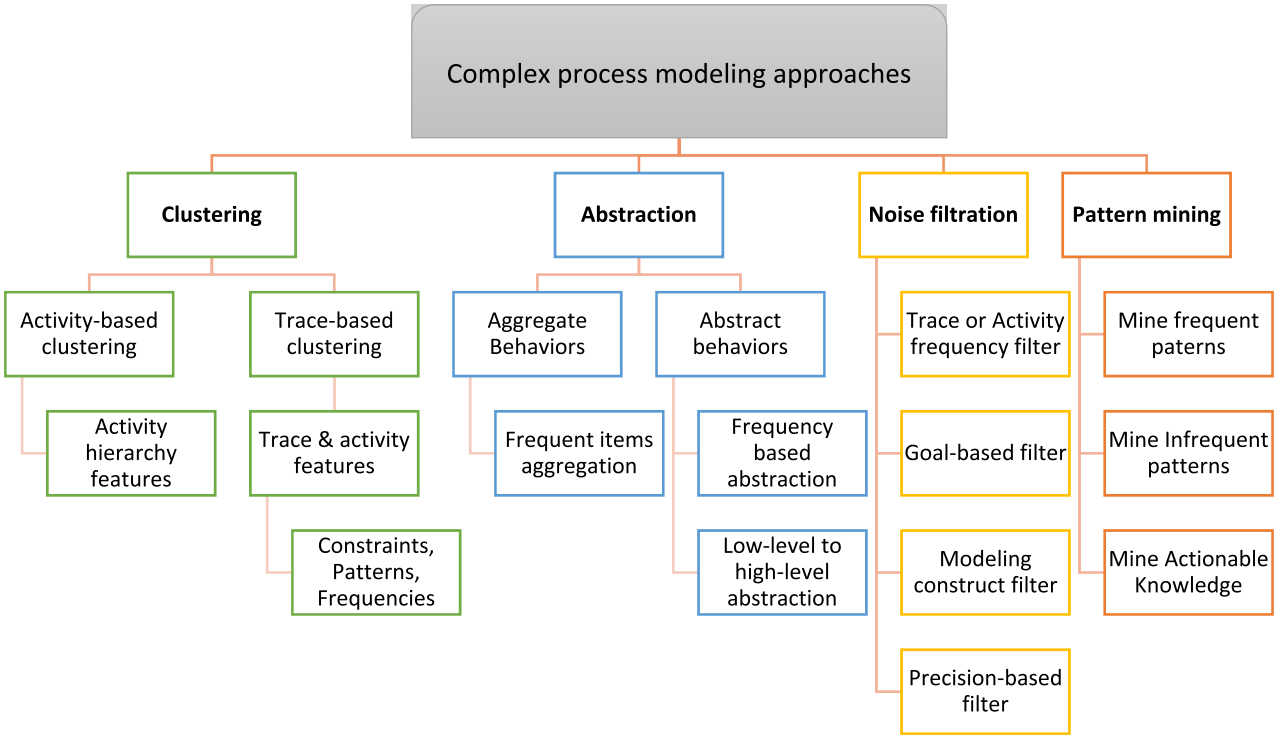

FIGURE 10. Taxonomy of Process complexity reduction approaches.

An important finding was that several studies judged model complexity just by visual analysis instead of any standard metric. At the same time, no justification was given for manual usage of a visual method for concluding about complexity. Nevertheless, the suitability of structural complexity measure is coupled with process modeling notation in hand.

C. RQ 3: WHAT ARE TECHNIQUES TO DEAL WITH PROCESS MODEL COMPLEXITY?

To answer this question, we first categorized the process complexity reduction techniques into approaches and sub-approaches. In the higher level, we classified the techniques according to the general approach they used, whereas in the subcategory, we further subclassified the general approaches according to the specific technique they used. Based on the results, we formulated the taxonomy of these approaches presented in Fig. 10. On the other hand, Fig. 11 reports the usage frequency of each approach calculated in percentages. The frequency is based on their usage in selected articles.

1) CLUSTERING-BASED APPROACHES

Clustering is the process of grouping items into similar containers known as Clusters. An event log may contain heterogeneous behavior, such as executing different sequences of activities in a process. Each process execution is called a process instance, also known as a Trace. In a sample trace <ABDCE>, Activity "A is directly followed by B" represents an instance of process execution behavior. Since traces may vary depending upon the execution behavior, the process model that originates from such an event log results in a so-called spaghetti-like representation leading to the complexity of the model [44]. The process of dividing the whole log into groups of Traces exhibiting similar behavior is called Trace clustering [43]. Some researchers also performed clustering to deal with fine-grained event logs, such as activity clustering [42], [43], [48], [49]. Based on the quantitative analysis, clustering in process mining remains the top choice for dealing with complexity. We categorize clustering into two subcategories, activity clustering and trace clustering.

Van Zelst and Cao [36], Song *et al.* [37], Delias *et al.* [42], and Evermann *et al.* [43] performed Trace segmentation to deal with fine-grained event log. They used the concept of co-occurrence of activities to cluster them together. Assy *et al.* [50] clustered traces based on a hierarchy of activities by grouping activities with a common label. Another work in this dimension was performed by Sun and Bauer [47], but their experiments were related to structured processes rather than unstructured processes. Van Zelst and Cao [36] presented the idea of clustering based upon the value of the attributes, but no experimental evaluation of the technique was performed. Another variant of clustering was by de Leoni and Dündar [51], who carried out a combination of abstraction and clustering. They identified batch sessions of events using the concept of the time interval, and the activities occurring together within short time intervals were clustered together to simplify models.

Chapela-Campa *et al.* [31] and Sun *et al.* [46], [47] used multiple features to cluster event logs. They presented the

concept of trace profiles as features of an event log, such as the frequency of a sequential relation A > B in an event log used as a feature to cluster event logs into homogenous sets. To some extent, they enhanced the behavioral quality of process models. In contrast, the process perspective, such as the complexity assessment of the process model, has largely been ignored, which is equally important. It remained unknown which features contribute to the optimal clustering in balancing behavioral quality and structural quality of process models. Here, the curse of dimensionality is yet another problem. In the presence of many features, finding an optimal feature for clustering is like finding a needle in a haystack. Nevertheless, feature selection techniques are potential approaches to investigate and find the solution to this problem. Moreover, the activity recurrence was also overlooked. The recursion can be considered a variable for complexity since recursion increases the number of arcs in the process model, which is one of the dominant factors in introducing complexity to process models, as seen in the previous section.

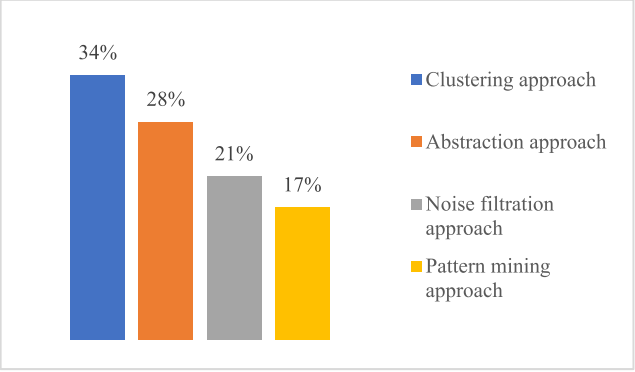

FIGURE 11. Usage frequency of complexity reduction approaches in selected articles.

In pattern-based and guided clustering approaches, Bose and Van Der Aalst [54] and Hompes *et al.* [55] performed clustering based upon specific patterns in traces that represent a deviation from the normal process execution. Similarly, Wang *et al.* [44] used a set of constraints on the clustering of traces; they first extracted the most prevalent process model from the process model and clustered traces according to this new model.

A guide clustering approach was adopted by Weerdt *et al.* [45]. They clustered traces using the so-called fitness measure of the process model to guide the clustering process. Process models having similar fitness were clustered together. Here it can be argued that the precision dimension is equally important as it restricts the modeling of additional behavior not seen in the log. Lu *et al.* [56] used a similar guided clustering technique based on process instance samples. Specifically, they used domain knowledge, e.g., sample process instances of diseases, and guided the clustering process based on those patterns. The traces in the process model having close precision to the provided patterns are clustered together.

Taking a different perspective on clustering approaches, De Koninck *et al.* [57] developed a technique to explain why certain traces were clustered into specific clusters. They did so by investigating common behavioral patterns in clusters, such as the presence of activity X and Y in a cluster or a relation X directly followed by Y in the cluster and many other such rules. Ekanayake *et al.* [40] used a mixed-method approach. They first clustered traces based upon variants, and then abstraction was introduced to derive subprocesses in each cluster by abstraction of activities that split and joined in the same place.

Conclusively, the clustering approaches effectively divide the log into subsets. Nevertheless, the main objective of reducing the complexity of process models and making them understandable has remained uninvestigated. The emphasis of evaluation mostly remained on the behavioral quality of process models. It is acknowledged that the resulting process model should exhibit similar behavior as in reference models, however, the actionable knowledge is also related to the simplicity of process models and is vital to consider [32], [58].

2) ABSTRACTION-BASED APPROACHES

In the process mining domain, Abstraction refers to hiding less critical information from the process model and showing them in an aggregate manner. The literature review revealed that the term ''abstraction'' remained synonymous with ''aggregation'' in process mining literature [59]. On the other hand, some researchers, Günther and Van Der Aalst [8] and Setiawan and Yahya [60], referred to aggregation as a specific kind of abstraction that shows behavior in an aggregate manner. The usage of abstraction-based techniques was found in a context where activities in a log contained a fine-grained level of information.

Günther and Van Der Aalst [8] first presented the concept of abstraction and aggregation in process mining. Their work is inspired by the idea of cartography (study of maps), where they aim to show only relevant information at a specific level. They used the concept of aggregation to show process model elements (arcs and nodes) in an aggregated manner while abstracting from insignificant details to simplify the model significantly. Instead of Petri net, they used Process maps, a Directly Follows Graph (DFG) based notation to represent the process model. Despite the inability of DFGs to differentiate between splits (AND & OR Splits), the DFG notation remains the most popular process modeling notation in process mining since 25 commercial process mining products use DFGs in their products [61]. However, their abstraction mechanism is guided by the frequency of activities, i.e., infrequent activities are abstracted from the process map. Despite simplifying the process models significantly, compliance checking is not trivial using such models.

a: LOW-LEVEL TO HIGH-LEVEL ABSTRACTION

The concept of holonym-meronymy (whole-part of) relation was used by Smirnov *et al.* [62] to introduce abstractions. They used a dictionary-based match of activities to infer the

relationship between two activities. The activities under the same holonym are abstracted, thus simplifying the process model. However, a dictionary-based activity hierarchy may not always represent a proper hierarchy. Context (domain semantics) of activity execution is essential here. Moreover, the behavioral or structural complexity measures were not assessed to validate model quality.

Deokar and Tao [63] and Smirnov *et al.* [64] used a similar concept of semantic relatedness. Instead of meronymy-based abstraction, they used the existing role hierarchy process model, and activities in the model were abstracted accordingly. However, their approach is led by an assumption that a process model containing process hierarchies is always present. Moreover, the problem of the same activity falling under multiple hierarchies was ignored. Another similar approach was suggested by Richetti *et al.* [59]. Their abstraction was limited to only one hierarchy level, and no evaluation was conducted against real-world datasets.

Baier *et al.* [28] and Ferreira *et al.* [65] introduced mapping of low-level activities to their high-level counterparts. The advantage over existing techniques was incorporating domain knowledge in the low-level to high-level activity mapping. A significant part of this mapping was carried out manually, and their work was not precise enough to map activities. Having low-level to high-level activity mapping in hand, Tax *et al.* [66] leveraged supervised learning techniques for process abstraction by using a sample of abstractions. However, their experiments were limited to simulated datasets only. Mannhardt *et al.* [67] used abstraction of low-level event logs to higher levels. Custom activity patterns and model-based activity patterns were used for the abstraction process. Their technique remained computationally expensive, and in the case of multiple abstraction candidates, only one was chosen arbitrarily with no validation of abstraction results.

Instead of using a single attribute for the abstraction of event log and activity hierarchy generation, Leemans *et al.* [41] proposed a multilevel activity hierarchy to simplify the model. Their technique accepts more than one attribute as a hierarchy classifier and uses them to perform multilevel abstractions according to the order of attributes.

Klessascheck *et al.* [68] considered the application of process mining on un-processed event logs as the reason for process complexity. They specifically focused on logs originating from the healthcare domain. Instead of specifying the name of the activity performed, only the drug names were used as the activity identifier. According to the context, they proposed activity transformation, such as specific medicine, that may belong to the NSAID category. Activity names were replaced with an abstracted version such as ''Prescribe NSAID.'' This way, multiple NSAID medicines were abstracted, resulting in a more straightforward process model. However, the quality of the abstraction remained unvalidated.

Li *et al.* [14] introduced the concept of classes to activities known as activity instances in process mining. Each activity containing the same preceding and succeeding activities is abstracted based on window size. They do so recursively by introducing multilevel abstractions over the log. Their technique simplifies the process model but is limited by assuming that each class at a specific abstraction level belongs to one higher level class. However, in real-life cases, the same class may belong to multiple higher-level process classes.

Instead of activities-based abstraction, Tsagkani and Tsalgatidou [38] considered additional attributes for abstractions, such as role hierarchies. However, their abstraction technique was highly influenced by the experience and expertise of those involved in abstraction.

b: FREQUENCY-BASED ABSTRACTION

Chapela-Campa *et al.* [31], [69] think that less frequent traces contribute to complexity in the process model, so the frequency-based abstraction of activities was carried out to simplify the model. The activities occurring under a specific threshold are abstracted and thus resulting in several process variants. They claimed their technique could simplify the process model in terms of complexity based on the P/T-CD metric, but no such results were demonstrated.

c: GOAL-BASED ABSTRACTION

Vathy-Fogarassy *et al.* [70] proposed a goal-based process discovery methodology specifically for a healthcare environment and ignored tasks not required to analyze disease under focus. They simplify the process model based on the domain-specific taxonomy by introducing multilevel abstractions over the log. However, their approach is specific to the healthcare domain where process taxonomy such as disease investigation levels is present. They used no behavioral or structural complexity metric for model assessment and relied on visual analysis. Nevertheless, their technique has the potential to be applied in other domains too. A similar healthcare specific approach was suggested by Erdogan and Tarhan [71]. Their approach remained focused on challenges associated with evaluation of complex processes primarily originating from healthcare domain.

3) NOISE FILTRATION APPROACHES

Noise filtration approaches apply filters on certain properties of log or model against specific thresholds to simplify the process model. No single definition of noise exists in process mining literature; instead, the noise is defined in an ad-hoc manner keeping in view the context. However, according to Sani *et al.* [9], filtering approaches lower the size of process instances needed by process discovery algorithms and thereby reducing the complexity of the process.

Considering the complex nature of processes in the Healthcare domain, Kaymak *et al.* [72] used a goal-based process discovery approach and filtered behavior irrelevant to the goal of process discovery. They claimed that their process is of medium complexity; however, they remained vague about complexity assessment and relied upon visual analysis of model complexity.

Weber *et al.* [73] attributed the ''split'' and ''join'' constructs in a process model as a reason for the noise and proposed that removing such constructs improved the understandability of the model. Cheng and Kumar [4] used the term ''log sanitization'' to perform noise filtration on the log. They cited duplicate, incomplete, inconsistent, and incorrect behavior as noise and proposed to remove such patterns from the log to improve process model comprehensibility. However, their definition of noise may have alternative explanations as well. Such as, they termed inconsistency in activity orders as noise; in reality, such patterns can result from compliance issues that require further investigation. Secondly, they did not evaluate their approach on real-world data sets.

De San Pedro *et al.* [74] relied on the behavioral precision metric to distinguish between noise and normal behavior and proposed simplifying logs by removing less-precise behavior. Although their approach is feasible for situations where precision is critical, still some level of generalization is necessary to retain the predictive power of the process model. Here, the notion of an incomplete log is also important if process data has been extracted from the process in execution; in such situations, the running cases will be removed due to low precision.

Conforti *et al.* [29] and Sani *et al.* [75] labeled infrequent behavior as noise and the reason for introducing additional arcs and nodes in the process model. They proposed to remove such behavior to simplify the process model. Notwithstanding, infrequent behavior is essential in terms of compliance perspective, and such behavior might also have alternative explanations, such as anomalous behavior that triggers the need for further investigation. A similar approach was used by Rashid *et al.* [76] to deal with complexity. They set a frequency-based threshold, removed infrequent behavior against it, and relied on visual analysis. As pointed out previously, the same argument of the importance of infrequent behavior holds for this study too.

An alternate terminology for noise, the chaotic activities, was introduced by Tax *et al.* [30]. They termed randomly occurring activities at different positions of the process model as chaotic activities. They argued that such random activities hinder the comprehension of process execution [30]. Although they successfully detected such patterns in the log, their impact on the complexity of the process model remained unevaluated, and no real-world explanation of such behavior was presented.

Vidgof *et al.* [77] considered the problem of removing infrequent behavior from the log. They proposed that rather than removing infrequent behavior, both the least frequent and most frequent behaviors should be preserved and included in the model. Their technique was an improvement over existing frequency-based noise filtration techniques in terms of considering both frequent and infrequent behavior. They relied on a visual analysis approach to assess complexity.

An incremental process discovery was proposed by Schuster *et al.* [78]. Rather than automatically discovering the whole process model, they suggest human involvement in discovering the process model and only model. Each trace is added to the process model, and its impact on model complexity is observed. The trace increasing model complexity is filtered out. The approach seems feasible when only a bunch of traces are mined. However, a high number of traces result in time-consuming process discovery and ignores the main objective of process mining, i.e., to discover the process automatically.

Finally, Zhang *et al.* [34] extracted mainstream behavior from the event log, i.e., the traces occurring more frequently or those containing frequently occurring activities. They used mainstream behaviors to extract behavioral probabilities of traces using Hidden Markov Models. The traces having less probability against mainstream behavior are removed from the log, thus simplifying the process model. Their frequency-based filtration technique also remained biased towards infrequent behavior.

Although filtration-based techniques use a straightforward method to deal with complexity, the frequency-based treatment of behavior filtration is somewhat unreasonable. From the perspective of compliance checking, the Infrequent behavior does not always represent noise [2]. Such behaviors are important for further investigation about why and when these happened. Conformance checking is a post-process discovery activity; this implies that if infrequent illegal behaviors are removed using filtration techniques, the violations against standard process executions would not be detected. This problem calls for techniques that can distinguish between ''infrequent legal behavior'' and ''infrequent illegal behavior'' and allow for filtration only over infrequent legal behavior to preserve the compliance checking properties in the log. Secondly, most noise filtration techniques rely on visual analysis for complexity analysis. The complexity analysis should be compared against some standard metrics. Thirdly there was a gap in evaluating techniques against real-world data sets and the context of real-world scenarios.

4) PATTERN-BASED APPROACHES

Pattern mining-based approaches simplify complex process models by extracting such execution behaviors from process models that contain specific patterns and only generate process models from such patterns. Such patterns represent subprocesses and process discovery based upon frequent behavioral patterns in the log, commonly referred to as Local Process Models (LPMs) [79]. LPM discovery techniques are guided by patterns. The traces which do not contain rule-satisfying patterns are removed from the log [80].

Frequency-based pattern mining approach was used by Liesaputra *et al.* [80]. They simplified the log based on the thresholds, such as the frequency of specific patterns in the log. Infrequent patterns are abstracted from, and only frequent patterns are mined.

Yahya *et al.* [58] introduced the actionable model discovery concept. They termed the actionable knowledge as behavioral patterns important to process analysts.

They proposed to apply constraints for retaining such behavior in the process model to reduce the number of patterns. Their technique remained subjective in defining actionable knowledge by domain experts rather than relying on model complexity measures. Moreover, the manual intervention of process analysts for marking actionable and non-actionable knowledge is time-consuming when a wide range of behaviors are available.

Diamantini *et al.* [81] proposed subprocess mining. They extracted all activity-department pairs occurring together from the log and introduced higher-level abstractions over these pairs. Then a process model is mined using the causal relations between two higher-level subprocesses. However, they did not assess their technique against the behavioral or structural complexity metrics.

The mining of Local Process Models was first proposed by Tax *et al.* [79]. They used well-known concepts of pattern coverage such as support and confidence metrics to mine frequent behavioral patterns from the log to extract the subsequence of patterns repeatedly appearing in the log. Their technique remained computationally expensive because many activities result in a high number of patterns. Tax *et al.* [32], [79] further extended their technique by taking care of the problem of a large number of patterns. They proposed considering only those patterns that provide some utility, i.e., only the interesting patterns in the mining context. They specified context-related constraints and applied them over patterns to retain only the concerning patterns. Similarly, Djenouri *et al.* [82] proposed the frequent itemset mining approach to deal with a large number of patterns by only considering the ones having a high support value.

Rather than focusing on frequent behavioral patterns, Chapela-Campa *et al.* [83] considered the irregular patterns important for analysis. They were the first to refrain from considering infrequent behavior as noise. They made a contrasting claim when compared with statements of other researchers. In comparison, other researchers argue that infrequent behaviors are one of the primary reasons for increased complexity. However, Chapela-Campa *et al.* [83] remained ambiguous on how mining less frequent behavioral patterns resolve the complexity problem. They also presented a frequent pattern mining approach [84]. Compared to existing techniques, their novelty was the ability to extract frequent structures such as loops, parallel, and selection structures.

In summary, the pattern mining approaches can greatly simplify process models, but only specific patterns or subprocesses are focused on rather than modeling all behavior. It was also observed they the pattern mining approaches remained computationally expensive. Further work on improving the computational expensiveness of such techniques is required. Secondly, the evaluation of such techniques was inspired by metrics from the data mining domain, such as support and confidence measures. No evaluation against structural complexity measures was performed to measure the complexity. Lastly, most of the pattern mining techniques work based on frequencies. In the case of highly variable data with low frequencies, it will be challenging to implement such techniques for model simplification.

In addition to the previously mentioned four dominant complexity reduction approaches, Kaouni *et al.* [85] proposed a visual analysis approach for complexity reduction. They propose using dotted charts and frequency-based graphs for process analysis. Although the graphs and dotted charts do help in conducting preliminary evaluation of the process data [2], [86], [87] and can be used as supplementary analysis types, however, the end-to-end analysis of the process is not possible using such visualizations.

Alongside the complexity reduction strategies, we also analyzed the literature on the types of datasets utilized for evaluation and validation of the complexity reduction techniques in order to determine the researchers' concentration on specific areas. Fig. 12 depicts the datasets of several processes utilized by researchers for evaluating and validating complexity reduction approaches. Almost one-third (30%) of the studies made use of healthcare datasets. The usage of datasets for the processes of securing a bank loan, managing incidents, and administering traffic fines were followed by healthcare domain. The majority of these datasets are from the Business Process Intelligence Challenge (BPIC) event, which is a business process analysis competition in which competitors get both real-world and synthetic datasets. In addition to process complexity research, these datasets are also popular and are commonly used throughout the process mining literature and openly accessible.

D. RQ 4: WHAT ARE THE STRENGTHS AND LIMITATIONS OF EXISTING TECHNIQUES DEALING WITH PROCESS MODEL COMPLEXITY?

This section presents the strength and limitations of each of the four approaches used for dealing with process complexity.

1) CLUSTERING-BASED APPROACHES

The strength of clustering-based techniques for clustering lies in dealing with process complexity in an unsupervised fashion and segmenting a complex process to traces level. However, this may also result in an unacceptable clustering solution when traces are clustered based on a specific perspective rather than random. Trace-level clustering approaches are suitable for dealing with model complexity when trace-level heterogeneity is observed in the log. In this direction, Jablonski *et al.* [52] used a frequency-based trace clustering method. Their clustering solution did not differentiate between frequent and infrequent behavioral patterns. Although there is a possibility that traces end up in the wrong clusters using their approach, nevertheless, all behavior is included in the final trace clustering solution. When trace level fitness is the goal, trace clustering is a better choice because it results in a good average fitness value [45].

The curse of dimensionality was another limitation in trace clustering approaches. When many features are available for clustering, dimensionality reduction techniques can be used, but this results in the loss of individual features' impact over

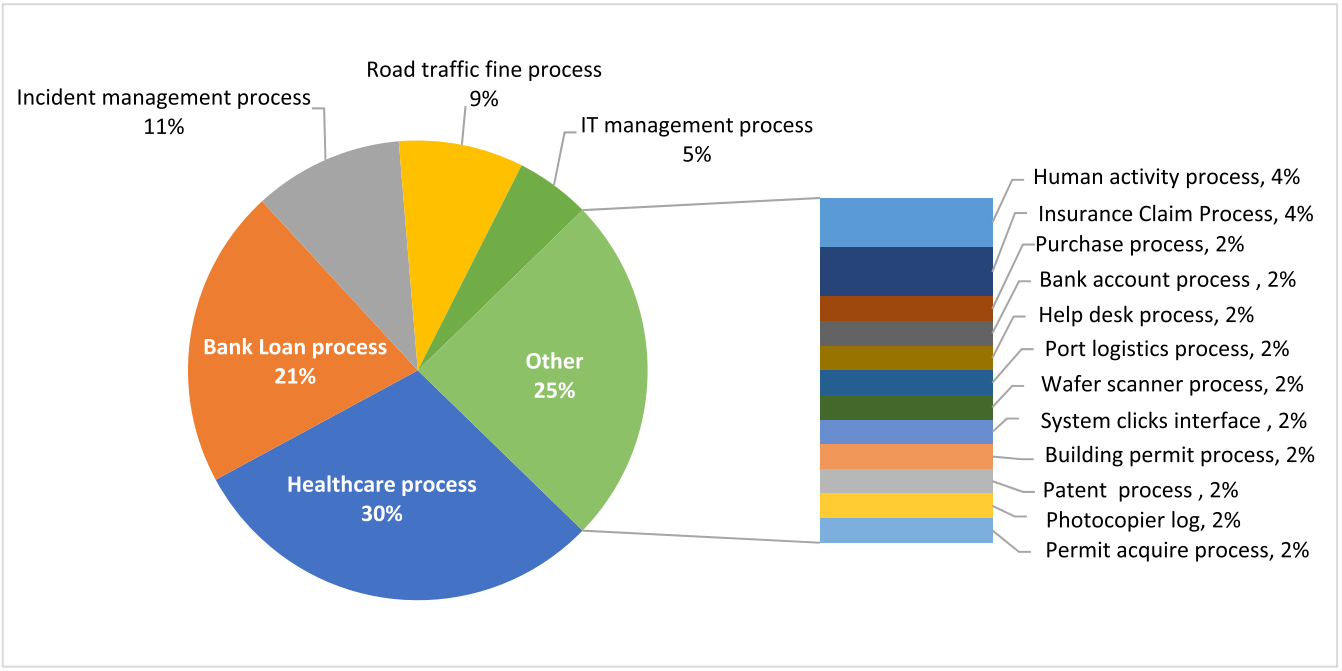

FIGURE 12. Datasets of different processes used for experiments and validation of complexity reduction techniques.

trace clustering. It was found that researchers have remained vague about an optimal number of clusters in terms of the striking best balance between behavioral and structural complexity measures. Model complexity is largely ignored during clustering, and only cluster homogeneity and behavioral quality measures are considered for measuring model complexity [36], [37], [52], [56]. Some researchers look at repeating trace behaviors in isolation [37], [52]. It can be argued that traces will end up in different clusters in frequency-based clustering if repeating behavioral patterns are also considered. So, the impact of repeating behaviors over clustering is worth the investigation.

2) ABSTRACTION-BASED APPROACHES

The strength of abstraction-based approaches lies in dealing with fine-grained process execution detail. Fine-grained refers to the log recording of each precise sub-step involved in process execution. The aim is to simplify process models by transforming a low-level log into a high-level counterpart. Such techniques are suitable for logs in which precise behaviors of process execution are found. Irrelevant behavior is abstracted from the final visualization; however, a clear distinction must be made between irrelevant and relevant behavior. It was noticed that researchers rarely consider matric-based complexity assessment. The process model evaluation was performed either using visual analysis [8], [68], [70] or behavioral complexity measures such as the fitness, precision & f-measure [31], [38], [62], [69].

Hierarchical abstraction has the potential to simplify the process model with customizability, but the presence of activity hierarchy is a prerequisite. Moreover, the same activity appearing in multiple levels of the activity hierarchy was used in isolation without considering the activity execution context to determine the correct level [14], [64], [67]. The validation of the abstraction solution remained missing and mostly remained influenced by the experience of the abstractor [38]. It was observed that experiments were performed on simulated datasets only, and patterns-based abstraction approaches remained expensive as it generates many candidate patterns for abstraction [67]. Further works on validation and optimization of abstraction-based approaches are required.

3) FILTRATION-BASED APPROACHES

As opposed to clustering and abstraction, the strength of filtration-based techniques lies in the straightforward treatment of the log to simplify the process model by applying constraints to remove noise from the log. Behaviors not fulfilling the pre-specified criteria are considered noise and thereby removed from the log. A frequency-based treatment was employed on the log to detect the noise and remove it. Some researchers used directly-follows dependency of events as a metric to filter logs from infrequent behaviors. In treating infrequent behavior, domain knowledge is of utmost importance as they are essential for compliance checking.

Similar to infrequent behaviors, the missing, swapped, or duplicate activities were also categorized as noise [4]. However, that can have alternate explanations, too, such as an indication of a potential compliance issue. Moreover, the behavioral precision-based variant of filtration techniques such as the one proposed by De San Pedro *et al.* [74] limits the showing the prediction behavior capability of the model.

All the above limitations must be considered when using filtration-based complexity approaches.

4) PATTERNS-BASED APPROACHES

The strength of pattern-based approaches lies in the flexibility to focus only on the intended part of the process [82]. Like abstraction-based approaches, patterns-based approaches also have good potential to deal with the fine-grained log. Researchers have primarily performed pattern mining based on the frequencies of the pattern [82], [88]. Guided patterns extraction is an important avenue to extract valid patterns, but no works were found in this direction. Another limitation was the problem of the computational expensiveness of pattern-based approaches [79] and found to be exponentially amplified with an increase in the number of activities [33]. Visual analysis was prevalently used for model complexity analysis by other approaches and was also found in pattern-based approaches [84]. Researchers remained focused on concepts from the data mining domain to validate patterns such as support, confidence [33], [79], and runtime performance measures. [81], [83]. It can be argued that pattern validation against the standard procedure is more important than frequency-based assessment.

E. RQ 5: WHAT ARE OPEN RESEARCH CHALLENGES AND AVENUES FOR FUTURE RESEARCH WORKS

Based on the shortcomings of the existing research, serval future research directions are proposed in this section to resolve unsolved challenges.

The clustering-based techniques remained the most significant in dealing with process complexity. In addition to behavioral complexity, there is a need for clustering approaches guided by model complexity rather than concluding about complexity at the end of the clustering process. When using such an approach, the primary goal of complexity reduction should be kept in mind. So far, it has been observed that clustering is performed by randomly selecting attributes at researchers' discretion. However, it is unclear which attributes contribute most to cluster homogeneity and improve model complexity during clustering.

Moreover, it has remained unclear how many clusters are enough to reduce model complexity. Every log has different properties, and based on log properties; different attributes can vary on cluster homogeneity, behavioral complexity, and structural complexity. In this direction, incorporating feature selection techniques is a potential avenue to select the best features that exhibit discriminative power. Finding and selecting the features that help reduce complexity for clustering solutions and striking an optimal balance between them is worth investigating. In most trace clustering techniques, a sequential flow between two activities is considered, although this can be justified because of having a process perspective. However, the effect of n-grams of sequential activities over cluster homogeneity in traces or partial traces is also worth investigating.

The abstraction approach combined with the clustering techniques is a prospective approach to better deal with model complexity. At the same time, trace clustering can also benefit from patterns-based approaches such as making clusters based on patterns rather than activity frequencies.

It was also noted that there is a lack of work on looking at model complexity from the log perspective. Similar labels are considered equal in clustering, but their context may differ [39]. Same activity labels from different departments affect the aggregate frequency and will be clustered together during clustering, but in reality, the context may differ, and they should be clustered separately. Moreover, in a manual recording of logs, an inconsistency in activity names will render an activity a separate modeling construct, thus increasing the complexity. The same holds for abstraction and patterns-based approaches, too, where the impact of duplicate instantiations of activity, e.g., duplicate label (same activity in multiple levels), needs to be evaluated. This calls for an investigation of the effect of such patterns on model complexity and the relevant remediation approaches.

One important finding was that many experiments had been conducted on already present datasets from the BPI Challenges data repository. One reason for this can be the unavailability of datasets. However, the datasets from other domains and sources must also be considered for the generalization of the approach.

Regarding the discriminatory treatment of infrequent behavior, there is a need to differentiate between real noise and infrequent behavior and redefinition of term noise in Process mining, such as differentiation of infrequent legal behavior and infrequent illegal behavior. There can be many alternative explanations for infrequent behavior in the log, e.g., infrequently occurring incompliant behavior is critical for compliance checking. An approach to clustering and modeling frequent and infrequent behaviors separately can help to reduce this impact. In a similar avenue, incorporating domain knowledge to filter out irrelevant activities is also a potential approach to deal with the effect of frequency-based biases towards logs to simplify the process model.

In the context of patterns-based approaches, the problem of computational complexity exists. Researchers encountered a high computation time during pattern detection [79]. Further research is required to optimize the pattern computation time.

Finally, it is recommended that the complexity metrics should be utilized for complexity assessment rather than relying on visual analysis. The type of modeling notation should also be kept in mind, as not all metrics work for all types of modeling notations. For example, the number of nodes and arcs, density, CNC, and CN measures seem equally feasible for two popular modeling notations, i.e., Petri net and Directly Follows Graphs (DFGs). On the other hand, P/T-CD and CFC metrics are unique to the Petri net modeling notation as they evaluate split constructs in the process model that are not used in Directly-Follows graphs.

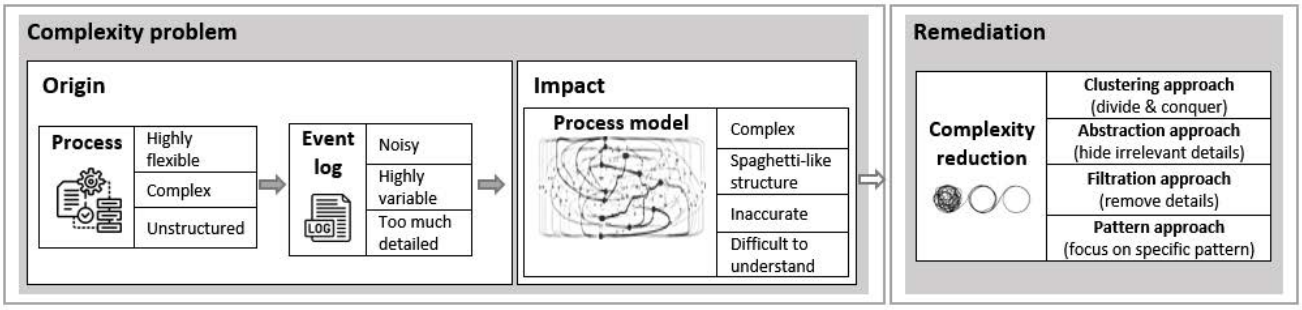

FIGURE 13. Conceptual model of Complexity problem in process mining.

IV. DISCUSSION

The results of the review indicate that the root cause of the complexity problem is the characteristics of the processes. Processes in some organizations are kept flexible or unstructured, which results in the event log being noisy, and significantly varying behavior is recorded in the log. However, this unstructuredness and variation affect process model quality, resulting in complex and spaghetti-like process models that are sometimes inaccurate. No actionable insights can be acquired from such models [14]. We have formulated a conceptual model of complex process modeling in process mining, as seen in Fig. 13. The grey part on the left-hand side of the model represents the complexity problem and its subparts, i.e., the origin of the problem and its impact. On the other hand, the remediation part of the model represents four prospective approaches as the strategies for resolving the complexity problem.

Although results indicate that the high flexibility and variation in process cause complexity in process models, sometimes, a slight variation in process execution also causes this problem. Based upon the context of the process execution, the position change of a single activity can also result in complexity and imprecision of process models. Tax *et al.* [30] have further researched this dimension, explicitly focusing on inaccuracy and complexity caused by an activity occurring at random positions in the process. Nevertheless, the questions like which activities are reasons for such issues, why this happens in the first place, and the impact of such disparity at different positions of process needs further investigation. In our view, the literature missed treatment of another significant cause of process complexity, i.e., the modeling of multiple subprocesses in a single model, which can also relate to the fine-grained level of process execution.

Taking a step back from post-processing approaches to deal with model complexity, if the process designers or the process managers can rethink the process flow and control the flexibility, the complexity problem will not occur in the first place. This seems feasible on the one hand, but it depends upon the rules and regulations of the organization since not all processes can be kept under strict constraints. Even if the flexibility is controlled, the fine-grained process execution is another bottleneck. Too much detail about process execution can be dealt with by only recording certain abstracted versions of activities. Nevertheless, the fine-grained recording of process execution is highly significant for diagnostics and accurately pinpointing process performance bottlenecks. Both the supervised and unsupervised event abstraction strategies are observed in the literature. Supervised abstraction is limited to being guided by domain knowledge, whereas in the case of unsupervised event abstraction, there remains uncertainty about whether the abstraction hierarchy is valid. It remains undiscovered how to assess the validity of abstraction in cases where no domain knowledge is present. Further research is needed in this direction.

In the case of post-event execution, the processes can be simplified using process mining algorithms. The researchers have proposed four approaches, clustering, abstraction, filtration, and pattern mining. The Fuzzy miner [8] incorporates clustering and abstraction mechanisms to simplify process models. Because of its simple, scalable, and filtered Directly-Follows Graphs (DFGs) based model generation, fuzzy miner remains the top choice for commercial process mining tools [61], [89]. However, DFG-based models fail to distinguish between a choice and a split construct [10].

The filtration approach was found to be used as an alternative choice. Frequency-based event prioritizing in the model is an interesting approach to only model certain recurring behaviors. Nevertheless, the filtration approach has negative implications for further advancing the process mining project. It threatens the validity of the conformance checking perspective of process mining, where process compliance is assessed against pre-specified rules and regulations. Incompliant behavior often infrequently occurs and will be ignored if filtration approaches are utilized. The filtration approaches can benefit from a prospective direction where a distinct treatment of frequent and infrequent behavior is made. Compliance-related significant behaviors should be preserved during the filtration process, and the rest of the behaviors can be filtered out as per normal flow.

Subjective process mining techniques, such as guided and Local Process Model (LPM) discovery, are bounded by the modeler's choice. The process modeler selects only those fragments of interest and concern to the process, e.g., process fragments that involve high financial costs. They are

superior to abstraction and filtration-based model discovery for simpler model production as their focus remains only on a specific part of the process regardless of frequent or infrequent behavior. However, LPM and guided process model discovery are only feasible when end-to-end model discovery is not a concern. Still, we think pattern-based approaches are potential approaches to dealing with logs where multiple processes or subprocesses are recorded in a single event log.

Results indicate that trace clustering remained the most popular technique for dealing with complexity. However, it was noted that rather than applying the straightforward simplification approach, the clustering techniques implicitly simplify process models by dividing the log into clusters based upon certain random features. Although, it is established that dividing the log into subsets improves individual subset complexity. However, it was unexpected to see that clustering was primarily carried out from a data perspective rather than a process perspective, i.e., whether the generated clusters are behaviorally correct. Considering the complexity perspective, it seems logical to let the clustering process be guided by features such as the complexity threshold of the models, where traces in each cluster are ranked according to their complexity. This can be achieved by using complexity measures and grouping the traces with a similar level of complexity into the same cluster according to the complexity threshold.

Further, categorizing the traces according to the complexity level will not be enough. Introducing abstraction-based techniques over clustering results can significantly result in an optimized solution. We call for further research to validate this mix and match combination.

The prevalence of healthcare datasets among those used for evaluation purposes demonstrates that the processes in the healthcare domain are more complex than any other domain. The patient treatment process in the healthcare industry consists of numerous activities and that too deals with various disciplines and subareas [11], [71], which creates inherent complexity. Similarly, the bank loan application process gets complex due to different process execution behaviors triggered by a range of customer types, loans, and financial statuses. Same can be stated for other evaluation datasets used throughout the complexity literature in process mining. On the other hand, the researchers' reliance on BPIC datasets can be attributed to the easy and open accessibility of these datasets. Nevertheless, there exists an empirical and knowledge gap on the usage of datasets self-collected by the researchers. Experiments conducted on self-collected datasets will help in revealing the other complexity perspectives such as complexity in data collection and preprocessing.

Results revealed random and subjective utilization of process complexity analysis metrics, mostly focusing on visual analysis of the model. Selection of a suitable process complexity metric is crucial since not all complexity quantification metrics apply in all scenarios. The appropriateness of complexity measure is coupled with process modeling notation in hand.

The findings of this review comprehensively shed light on the process complexity problem in process mining. We think that the arguments about practical and theoretical implications of existing approaches will help the novice and the currently working researchers in this domain understand this problem in a broader context. Moreover, it is expected to pave the way for extending knowledge in this direction using proposed future research endeavors.

Although this review was intended to be comprehensive, there are some threats to the validity of the results and findings. For quality purposes, the papers indexed in popular databases having at least one citation were included in the review. Ignoring this criterion can expand the number of articles; however, this compromises the quality. Moreover, the focus remained on journal articles and conference proceedings. The book chapters and workshop papers were excluded. However, many conference proceedings have been published as book chapters in the well-known Springer Lecture Notes Series. Such articles are unaffected by our book exclusion criteria.

V. CONCLUSION

Process mining techniques hold the potential to find bottlenecks and improve the business processes of organizations. However, If the process mining results are not understandable or complex, the whole project becomes useless. Several researchers approached the complexity problem in process mining, but a general overview of the topic at hand remained missing. In this paper, we conducted a systematic literature review to have a unified overview of the approaches for dealing with the problem of complexity in the process mining domain. Six well-known research databases were searched. In addition to formulating a conceptual model of complexity problem, a taxonomy of complexity reduction approaches was also formulated. It was identified how the process complexity problem is realized across different studies, what factors contribute to it, and how complexity is analyzed and prevented. Subsequently, the identification of research gaps and future research directions are proposed.

Findings reveal that the flexibility in the process, the fine-grained level of detail, and noise in the logs are the main contributors to process complexity. Moreover, it was found that the complexity problem is solved using four prospective approaches, clustering, abstraction, noise removal, and patterns mining. Different metrics used for these measures were identified. It was also noted that the emphasis of complexity analysis remained on behavioral complexity measures. At the same time, less importance is given to structural complexity, which directly relates to the process model's comprehensibility. Finally, several research gaps and future research directions are also presented.

REFERENCES

[1] A. Bolt, M. de Leoni, and W. M. P. van der Aalst, "Scientific workflows for process mining: Building blocks, scenarios, and implementation," *Int. J. Softw. Tools Technol. Transf.*, vol. 18, no. 6, pp. 607–628, Nov. 2016, doi: 10.1007/S10009-015-0399-5.

[2] W. van der Aalst, *Data Science in Action*. Berlin, Germany: Springer, 2016.
[3] C. Ortmeier, N. Henningsen, A. Langer, A. Reiswich, A. Karl, and C. Herrmann, "Framework for the integration of process mining into life cycle assessment," *Proc. CIRP*, vol. 98, pp. 163–168, Jan. 2021, doi: 10.1016/J.PROCIR.2021.01.024.
[4] H.-J. Cheng and A. Kumar, "Process mining on noisy logs—Can log sanitization help to improve performance?" *Decis. Support Syst.*, vol. 79, pp. 138–149, Nov. 2015, doi: 10.1016/j.dss.2015.08.003.
[5] C. D. S. Garcia, A. Meincheim, E. R. F. Junior, M. R. Dallagassa, D. M. V. Sato, D. R. Carvalho, E. A. P. Santos, and E. E. Scalabrin, "Process mining techniques and applications—A systematic mapping study," *Expert Syst. Appl.*, vol. 133, pp. 260–295, Nov. 2019, doi: 10.1016/j.eswa.2019.05.003.
[6] A. J. M. M. Weijters, W. M. P. van der Aalst, and A. K. A. de Medeiros, "Process mining with the heuristics miner-algorithm," in *BETA Publicatie: Working Papers*. Eindhoven, The Netherlands: Technische Universiteit Eindhoven, 2006.
[7] S. J. J. Leemans, D. Fahland, and W. M. P. van der Aalst, "Discovering block-structured process models from incomplete event logs," in *Proc. Int. Conf. Appl. Theory Petri Nets Concurrency*, 2014, pp. 91–110, doi: 10.1007/978-3-319-07734-5_6.
[8] C. W. Günther and W. M. P. Van Der Aalst, "Fuzzy mining—Adaptive process simplification based on multi-perspective metrics," in *Business Process Management* (Lecture Notes in Computer Science) vol. 4714. Berlin, Germany: Springer, 2007, pp. 328–343, doi: 10.1007/978-3-540-75183-0_24.
[9] M. F. Sani, S. J. van Zelst, and W. M. P. van der Aalst, "The impact of biased sampling of event logs on the performance of process discovery," *Computing*, vol. 103, no. 6, pp. 1085–1104, Jun. 2021, doi: 10.1007/s00607-021-00910-4.
[10] Z. Lamghari, M. Radgui, R. Saidi, and M. D. Rahmani, "An operational support approach for mining unstructured business processes," *Revista de Informática Teórica e Aplicada*, vol. 28, no. 1, pp. 22–38, Jan. 2021, doi: 10.22456/2175-2745.106277.
[11] A. F. D. Gomes, A. C. W. G. de Lacerda, and J. R. da Silva Fialho, "Comparative analysis of process mining algorithms in Python," in *Proc. Int. Conf. Smart Objects Technol. Social Good*, vol. 401, 2021, pp. 27–43, doi: 10.1007/978-3-030-91421-9_3.
[12] J. Mendling, H. A. Reijers, and J. Cardoso, "What makes process models understandable?" in *Proc. Int. Conf. Bus. Process Manag.*, vol. 4714, 2007, pp. 48–63, doi: 10.1007/978-3-540-75183-0_4.
[13] H. A. Reijers and J. Mendling, "A study into the factors that influence the understandability of business process models," *IEEE Trans. Syst., Man, Cybern. A, Syst. Humans*, vol. 41, no. 3, pp. 449–462, May 2011, doi: 10.1109/TSMCA.2010.2087017.
[14] C.-Y. Li, S. J. van Zelst, and W. M. P. van der Aalst, "An activity instance based hierarchical framework for event abstraction," in *Proc. 3rd Int. Conf. Process Mining (ICPM)*, Oct. 2021, pp. 160–167, doi: 10.1109/ICPM53251.2021.9576868.
[15] S. Ya'acob, N. M. Ali, N. M. Nayan, H.-N. Liang, I. Ahmad, R. Ibrahim, and N. A. A. Bakar, "Visual analytics evaluation process: Practice guidelines for complex domain," *Malaysian J. Comput. Sci.*, pp. 118–134, Nov. 2019, doi: 10.22452/mjcs.sp2019no1.9.
[16] B. Kitchenham, O. P. Brereton, D. Budgen, M. Turner, J. Bailey, and S. Linkman, "Systematic literature reviews in software engineering—A systematic literature review," *Inf. Softw. Technol.*, vol. 51, no. 1, pp. 7–15, Jan. 2009, doi: 10.1016/j.infsof.2008.09.009.
[17] M. La Rosa, P. Wohed, J. Mendling, A. H. M. ter Hofstede, H. A. Reijers, and W. M. P. van der Aalst, "Managing process model complexity via abstract syntax modifications," *IEEE Trans. Ind. Informat.*, vol. 7, no. 4, pp. 614–629, Nov. 2011, doi: 10.1109/TII.2011.2166795.
[18] H. Schonenberg, R. Mans, N. Russell, N. Mulyar, and W. van der Aalst, "Process flexibility: A survey of contemporary approaches," in *Advances in Enterprise Engineering I* (Lecture Notes in Business Information Processing), vol. 10. Berlin, Germany: Springer, 2008, pp. 16–30, doi: 10.1007/978-3-540-68644-6_2.
[19] C. Houy, P. Fettke, P. Loos, W. M. P. Van Der Aalst, and J. Krogstie, "BPM-in-the-large—Towards a higher level of abstraction in business process management," in *IFIP Advances in Information and Communication Technology*, vol. 334. Berlin, Germany: Springer, 2010, pp. 233–244.
[20] R. J. D'Castro, A. L. I. Oliveira, and A. H. Terra, "Process mining discovery techniques in a low-structured process works?" in *Proc. 7th Brazilian Conf. Intell. Syst. (BRACIS)*, Oct. 2018, pp. 200–205, doi: 10.1109/BRACIS.2018.00042.
[21] C. Duan and Q. Wei, "Process mining of duplicate tasks: A systematic literature review," in *Proc. IEEE Int. Conf. Artif. Intell. Comput. Appl. (ICAICA)*, Jun. 2020, pp. 778–784, doi: 10.1109/ICAICA50127.2020.9182667.
[22] S. J. van Zelst, F. Mannhardt, M. de Leoni, and A. Koschmider, "Event abstraction in process mining: Literature review and taxonomy," *Granular Comput.*, vol. 6, no. 3, pp. 719–736, Jul. 2021, doi: 10.1007/s41066-020-00226-2.
[23] M. Gusenbauer and N. R. Haddaway, "Which academic search systems are suitable for systematic reviews or meta-analyses? Evaluating retrieval qualities of Google scholar, PubMed, and 26 other resources," *Res. Synth. Methods*, vol. 11, no. 2, pp. 181–217, Mar. 2020, doi: 10.1002/jrsm.1378.
[24] A. W. Harzing. (2007). *Publish or Perish*. Accessed: Aug. 10, 2022. [Online]. Available: https://harzing.com/resources/publish-or-perish
[25] M. J. Page, J. E. McKenzie, P. M. Bossuyt, I. Boutron, T. C. Hoffmann, C. D. Mulrow, L. Shamseer, J. M. Tetzlaff, E. A. Akl, S. E. Brennan, and R. Chou, "The PRISMA 2020 statement: An updated guideline for reporting systematic reviews," *Brit. Med. J.*, vol. 372, p. 71, Mar. 2021, doi: 10.1136/bmj.n71.
[26] M. J. Page, D. Moher, P. M. Bossuyt, I. Boutron, T. C. Hoffmann, C. D. Mulrow, L. Shamseer, J. M. Tetzlaff, E. A. Akl, S. E. Brennan, and R. Chou, "PRISMA 2020 explanation and elaboration: Updated guidance and exemplars for reporting systematic reviews," *Brit. Med. J.*, vol. 372, p. 160, Mar. 2021, doi: 10.1136/bmj.n160.
[27] N. S. N. Ayutaya, P. Palungsuntikul, and W. Premchaiswadi, "Heuristic mining: Adaptive process simplification in education," in *Proc. 10th Int. Conf. ICT Knowl. Eng.*, Nov. 2012, pp. 221–227, doi: 10.1109/ICTKE.2012.6408559.
[28] T. Baier, J. Mendling, and M. Weske, "Bridging abstraction layers in process mining," *Inf. Syst.*, vol. 46, pp. 123–139, Dec. 2014, doi: 10.1016/j.is.2014.04.004.
[29] R. Conforti, M. L. Rosa, and A. H. T. Hofstede, "Filtering out infrequent behavior from business process event logs," *IEEE Trans. Knowl. Data Eng.*, vol. 29, no. 2, pp. 300–314, Feb. 2017, doi: 10.1109/TKDE.2016.2614680.
[30] N. Tax, N. Sidorova, and W. M. P. van der Aalst, "Discovering more precise process models from event logs by filtering out chaotic activities," *J. Intell. Inf. Syst.*, vol. 52, no. 1, pp. 107–139, Feb. 2019, doi: 10.1007/s10844-018-0507-6.
[31] D. Chapela-Campa, M. Mucientes, and M. Lama, "Simplification of complex process models by abstracting infrequent behaviour," in *Service-Oriented Computing* (Lecture Notes in Computer Science), vol. 11895. Cham, Switzerland: Springer, Oct. 2019, pp. 415–430, doi: 10.1007/978-3-030-33702-5_32.
[32] N. Tax, B. Dalmas, N. Sidorova, W. M. P. van der Aalst, and S. Norre, "Interest-driven discovery of local process models," *Inf. Syst.*, vol. 77, pp. 105–117, Sep. 2018, doi: 10.1016/J.IS.2018.04.006.
[33] N. Tax, N. Sidorova, W. M. P. van der Aalst, and R. Haakma, "LocalProcessModelDiscovery: Bringing Petri nets to the pattern mining world," in *Proc. Int. Conf. Appl. Theor. Petri Nets Concurrency*, vol. 10877, Jun. 2018, pp. 374–384, doi: 10.1007/978-3-319-91268-4_20.
[34] Z. Zhang, R. Hildebrant, F. Asgarinejad, N. Venkatasubramanian, and S. Ren, "Improving process discovery results by filtering out outliers from event logs with hidden Markov models," in *Proc. IEEE 23rd Conf. Bus. Informat. (CBI)*, Sep. 2021, pp. 171–180, doi: 10.1109/CBI52690.2021.00028.
[35] M. Lall, J. A. Van Der Poll, and L. M. Venter, "A process model for the formalization of quality attributes of service-based software systems," *Malaysian J. Comput. Sci.*, vol. 32, no. 4, pp. 284–303, Oct. 2019, doi: 10.22452/MJCS.VOL32NO4.3.
[36] S. J. Van Zelst and Y. Cao, "A generic framework for attribute-driven hierarchical trace clustering," in *Proc. Int. Conf. Bus. Process Manag.*, 2020, pp. 308–320.
[37] M. Song, H. Yang, S. H. Siadat, and M. Pechenizkiy, "A comparative study of dimensionality reduction techniques to enhance trace clustering performances," *Expert Syst. Appl.*, vol. 40, no. 9, pp. 3722–3737, Jul. 2013, doi: 10.1016/j.eswa.2012.12.078.
[38] C. Tsagkani and A. Tsalgatidou, "Process model abstraction for rapid comprehension of complex business processes," *Inf. Syst.*, vol. 103, Jan. 2022, Art. no. 101818, doi: 10.1016/j.is.2021.101818.
[39] N. Wang, S. Sun, and D. Ouyang, "Business process modeling abstraction based on semi-supervised clustering analysis," *Bus. Inf. Syst. Eng.*, vol. 60, no. 6, pp. 525–542, Dec. 2018, doi: 10.1007/s12599-016-0457-x.

[40] C. C. Ekanayake, M. Dumas, L. García-Bañuelos, and M. La Rosa, "Slice, mine and dice: Complexity-aware automated discovery of business process models," in *Proc. 11th Int. Conf. Bus. Process Manag.*, vol. 8094, 2013, pp. 49–64, doi: 10.1007/978-3-642-40176-3_6.

[41] S. J. J. Leemans, K. Goel, and S. J. van Zelst, "Using multi-level information in hierarchical process mining: Balancing behavioural quality and model complexity," in *Proc. 2nd Int. Conf. Process Mining (ICPM)*, Oct. 2020, pp. 137–144, doi: 10.1109/ICPM49681.2020.00029.

[42] P. Delias, M. Doumpos, E. Grigoroudis, P. Manolitzas, and N. Matsatsinis, "Supporting healthcare management decisions via robust clustering of event logs," *Knowl.-Based Syst.*, vol. 84, pp. 203–213, Aug. 2015, doi: 10.1016/j.knosys.2015.04.012.

[43] J. Evermann, T. Thaler, and P. Fettke, "Clustering traces using sequence alignment," in *Proc. Int. Conf. Bus. Process Manag.*, vol. 256, 2016, pp. 179–190, doi: 10.1007/978-3-319-42887-1_15.

[44] P. Wang, W. Tan, A. Tang, and K. Hu, "A novel trace clustering technique based on constrained trace alignment," in *Proc. Int. Conf. Hum. Centered Comput.*, 2018, pp. 53–63, doi: 10.1007/978-3-319-74521-3_7.

[45] J. De Weerdt, S. K. L. M. vanden Broucke, J. Vanthienen, and B. Baesens, "Active trace clustering for improved process discovery," *IEEE Trans. Knowl. Data Eng.*, vol. 25, no. 12, pp. 2708–2720, Dec. 2013, doi: 10.1109/TKDE.2013.64.

[46] Y. Sun, B. Bauer, and M. Weidlich, "Compound trace clustering to generate accurate and simple sub-process models," in *Proc. Int. Conf. Serv. Comput.*, vol. 10601, Nov. 2017, pp. 175–190, doi: 10.1007/978-3-319-69035-3_12.

[47] Y. Sun and B. Bauer, "A novel top-down approach for clustering traces," in *Proc. Int. Conf. Adv. Inf. Syst. Eng.*, vol. 9097, 2015, pp. 331–345, doi: 10.1007/978-3-319-19069-3_21.

[48] C. W. Günther, A. Rozinat, and W. M. P. Van Der Aalst, "Activity mining by global trace segmentation," in *Proc. Int. Conf. Bus. Process Manag.*, 2010, pp. 128–139, doi: 10.1007/978-3-642-12186-9_13.

[49] G. M. Veiga and D. R. Ferreira, "Understanding spaghetti models with sequence clustering for ProM," in *Proc. Int. Conf. Bus. Process Manag.*, vol. 43, 2010, pp. 92–103, doi: 10.1007/978-3-642-12186-9_10.

[50] N. Assy, B. F. van Dongen, and W. M. P. van der Aalst, "Discovering hierarchical consolidated models from process families," in *Advanced Information Systems Engineering* (Lecture Notes in Computer Science), vol. 10253. Cham, Switzerland: Springer, 2017, pp. 314–329.

[51] M. de Leoni and S. Dündar, "Event-log abstraction using batch session identification and clustering," in *Proc. 35th Annu. ACM Symp. Appl. Comput.*, Mar. 2020, pp. 36–44, doi: 10.1145/3341105.3373861.

[52] S. Jablonski, M. Röglinger, S. Schönig, and K. Wyrtki, "Multi-perspective clustering of process execution traces," *Enterp. Model. Inf. Syst. Archit.*, vol. 14, no. 2, pp. 1–22, 2019, doi: 10.18417/emisa.14.2.

[53] M. Song, C. W. Gunther, and W. M. P. van der Aalst, "Trace clustering in process mining," in *Proc. Int. Conf. Bus. Process Manag.*, vol. 17, 2009, pp. 109–120.

[54] R. P. Bose and W. M. P. Van Der Aalst, "Trace clustering based on conserved patterns: Towards achieving better process models," in *Business Process Management Workshops* (Lecture Notes in Business Information Processing), vol. 43. Berlin, Germany: Springer, 2009, pp. 170–181.

[55] B. Hompes, J. Buijs, W. Aalst, P. Dixit, and J. Buurman, "Discovering deviating cases and process variants using trace clustering," in *Proc. Benelux Conf. Artif. Intell. (BNAIC)*, 2015, pp. 8–17.

[56] X. Lu, S. A. Tabatabaei, M. Hoogendoorn, and H. A. Reijers, "Trace clustering on very large event data in healthcare using frequent sequence patterns," in *Business Process Management* (Lecture Notes in Computer Science), vol. 11675. Berlin, Germany: Springer, 2019, pp. 198–215.

[57] P. De Koninck, J. De Weerdt, and S. K. L. M. van den Broucke, "Explaining clusterings of process instances," *Data Mining Knowl. Discovery*, vol. 31, no. 3, pp. 774–808, May 2017, doi: 10.1007/s10618-016-0488-4.

[58] B. N. Yahya, M. Song, H. Bae, S. Sul, and J.-Z. Wu, "Domain-driven actionable process model discovery," *Comput. Ind. Eng.*, vol. 99, pp. 382–400, Sep. 2016, doi: 10.1016/j.cie.2016.05.010.

[59] P. H. P. Richetti, F. A. Baião, and F. M. Santoro, "Declarative process mining: Reducing discovered models complexity by pre-processing event logs," in *Business Process Management* (Lecture Notes in Computer Science), vol. 8659. Cham, Switzerland: Springer, 2014, pp. 400–407.

[60] F. Setiawan and B. N. Yahya, "Improved behavior model based on sequential rule mining," *Appl. Soft Comput.*, vol. 68, pp. 944–960, Jul. 2018, doi: 10.1016/j.asoc.2018.01.035.

[61] W. M. P. van der Aalst, "Process discovery from event data: Relating models and logs through abstractions," *WIREs Data Mining Knowl. Discovery*, vol. 8, no. 3, p. e1244, May 2018, doi: 10.1002/widm.1244.

[62] S. Smirnov, R. Dijkman, J. Mendling, and M. Weske, "Meronymy-based aggregation of activities in business process models," in *Conceptual Modeling*. Berlin, Germany: Springer, 2010, pp. 1–14.

[63] A. V. Deokar and J. Tao, "Semantics-based event log aggregation for process mining and analytics," *Inf. Syst. Frontiers*, vol. 17, no. 6, pp. 1209–1226, Dec. 2015, doi: 10.1007/s10796-015-9563-4.

[64] S. Smirnov, H. A. Reijers, and M. Weske, "A semantic approach for business process model abstraction," in *Proc. Int. Conf. Adv. Inf. Syst. Eng. (CAiSE)*, vol. 6741, 2011, pp. 497–511, doi: 10.1007/978-3-642-21640-4_37.

[65] D. R. Ferreira, F. Szimanski, and C. G. Ralha, "Improving process models by mining mappings of low-level events to high-level activities," *J. Intell. Inf. Syst.*, vol. 43, no. 2, pp. 379–407, Oct. 2014, doi: 10.1007/s10844-014-0327-2.

[66] N. Tax, N. Sidorova, R. Haakma, and W. M. P. Van Der Aalst, "Event abstraction for process mining using supervised learning techniques," in *Proc. of SAI Intell. Syst. Conf. (IntelliSys)*, 2016, pp. 251–269, doi: 10.1007/978-3-319-56994-9_18.

[67] F. Mannhardt, M. de Leoni, H. A. Reijers, W. M. P. V. D. Aalst, and P. J. Toussaint, "Guided process discovery—A pattern-based approach," *Inf. Syst.*, vol. 76, pp. 1–18, Jul. 2018, doi: 10.1016/j.is.2018.01.009.

[68] F. Klessascheck, T. Lichtenstein, M. Meier, S. Remy, J. P. Sachs, L. Pufahl, R. Miotto, E. Boettinger, and M. Weske, "Domain-specific event abstraction," *Bus. Inf. Syst.*, vol. 2021, pp. 117–126, Jul. 2021, doi: 10.52825/BIS.V1I.39.

[69] D. Chapela-Campa, M. Mucientes, and M. Lama, "Understanding complex process models by abstracting infrequent behavior," *Future Gener. Comput. Syst.*, vol. 113, pp. 428–440, Dec. 2020, doi: 10.1016/j.future.2020.07.030.

[70] Á. Vathy-Fogarassy, I. Vassányi, and I. Kósa, "Multi-level process mining methodology for exploring disease-specific care processes," *J. Biomed. Informat.*, vol. 125, Jan. 2022, Art. no. 103979, doi: 10.1016/j.jbi.2021.103979.

[71] T. G. Erdogan and A. Tarhan, "A goal-driven evaluation method based on process mining for healthcare processes," *Appl. Sci.*, vol. 8, no. 6, p. 894, May 2018, doi: 10.3390/APP8060894.

[72] U. Kaymak, R. Mans, T. V. D. Steeg, and M. Dierks, "On process mining in health care," in *Proc. IEEE Int. Conf. Syst., Man, Cybern. (SMC)*, Oct. 2012, pp. 1859–1864, doi: 10.1109/ICSMC.2012.6378009.

[73] P. Weber, B. Bordbar, and P. Tino, "A principled approach to mining from noisy logs using Heuristics Miner," in *Proc. IEEE Symp. Comput. Intell. Data Mining*, Apr. 2013, pp. 119–126, doi: 10.1109/CIDM.2013.6597226.

[74] J. De San Pedro, J. Carmona, and J. Cortadella, "Log-based simplification of process models," in *Proc. Int. Conf. Bus. Process Manag.*, vol. 9253, 2015, pp. 457–474, doi: 10.1007/978-3-319-23063-4_30.

[75] M. F. Sani, S. J. van Zelst, and W. M. P. van der Aalst, "Improving process discovery results by filtering outliers using conditional behavioural probabilities," in *Proc. Bus. Process Manage. Workshops*, vol. 308, 2017, pp. 216–229, doi: 10.1007/978-3-319-74030-0_16.

[76] M. Rashid, H. Naeem, M. Aamir, W. Ali, and W. Ahmed, "A multi-level process mining framework for correlating and clustering of biomedical activities using event logs," *Int. J. Adv. Comput. Sci. Appl.*, vol. 8, no. 3, pp. 393–401, 2017, doi: 10.14569/IJACSA.2017.080354.

[77] M. Vidgof, D. Djurica, S. Bala, and J. Mendling, "Cherry-picking from spaghetti: Multi-range filtering of event logs," in *Enterprise, Business-Process and Information Systems Modeling* (Lecture Notes in Business Information Processing), vol. 387. Cham, Switzerland: Springer, 2020, pp. 135–149.

[78] D. Schuster, S. J. van Zelst, and W. M. P. van der Aalst, "Incremental discovery of hierarchical process models," in *Research Challenges in Information Science* (Lecture Notes in Business Information Processing), vol. 385. Cham, Switzerland: Springer, 2020, pp. 417–433.

[79] N. Tax, N. Sidorova, R. Haakma, and W. M. P. van der Aalst, "Mining local process models," *J. Innov. Digit. Ecosyst.*, vol. 3, no. 2, pp. 183–196, Dec. 2016, doi: 10.1016/j.jides.2016.11.001.

[80] V. Liesaputra, S. Yongchareon, and S. Chaisiri, "Efficient process model discovery using maximal pattern mining," in *Proc. Int. Conf. Bus. Process Manag.* Cham, Switzerland: Springer, 2015, pp. 441–456.

[81] C. Diamantini, L. Genga, and D. Potena, "Behavioral process mining for unstructured processes," *J. Intell. Inf. Syst.*, vol. 47, no. 1, pp. 5–32, Aug. 2016, doi: 10.1007/s10844-016-0394-7.
[82] Y. Djenouri, A. Belhadi, and P. Fournier-Viger, "Extracting useful knowledge from event logs: A frequent itemset mining approach," *Knowl.-Based Syst.*, vol. 139, pp. 132–148, Jan. 2018, doi: 10.1016/j.knosys.2017.10.016.
[83] D. Chapela-Campa, M. Mucientes, and M. Lama, "Discovering infrequent behavioral patterns in process models," in *Proc. Int. Conf. Bus. Process Manag.* Cham, Switzerland: Springer, 2017, pp. 324–340.
[84] D. Chapela-Campa, M. Mucientes, and M. Lama, "Mining frequent patterns in process models," *Inf. Sci.*, vol. 472, pp. 235–257, Jan. 2019, doi: 10.1016/j.ins.2018.09.011.
[85] A. Kaouni, G. Theodoropoulou, A. Bousdekis, A. Voulodimos, and G. Miaoulis, "Visual analytics in process mining for supporting business process improvement," *Front. Artif. Intell. Appl.*, vol. 338, p. 5, Oct. 2021, doi: 10.3233/FAIA210089.
[86] D. R. Ferreira, *A Primer on Process Mining*. Cham, Switzerland: Springer, 2017.
[87] J.-F. Rodríguez-Quintero, A. Sánchez-Díaz, L. Iriarte-Navarro, A. Maté, M. Marco-Such, and J. Trujillo, "Fraud audit based on visual analysis: A process mining approach," *Appl. Sci.*, vol. 11, no. 11, p. 4751, May 2021, doi: 10.3390/app11114751.
[88] M. V. M. Kumar, L. Thomas, and B. Annappa, "Simplifying spaghetti processes to find the frequent execution paths," in *Proc. 1st Int. Conf. Smart Syst., Innov. Comput.*, vol. 79, 2018, pp. 693–701, doi: 10.1007/978-981-10-5828-8_66.
[89] M. R. Dallagassa, C. dos Santos Garcia, E. E. Scalabrin, S. O. Ioshii, and D. R. Carvalho, "Opportunities and challenges for applying process mining in healthcare: A systematic mapping study," *J. Ambient Intell. Hum. Comput.*, vol. 13, no. 1, pp. 165–182, Jan. 2022, doi: 10.1007/s12652-021-02894-7.

MOHAMMAD IMRAN (Member, IEEE) received the B.S. degree in information technology from the University of Balochistan, in 2013, and the M.S. degree in information technology from the Balochistan University of Information Technology, Engineering and Management Sciences (BUITEMS), in 2017. He is currently pursuing the Ph.D. degree in information systems with FSKTM, University of Malaya, Malaysia. His current research interests include business process intelligence, business process management, and process mining domain with a special focus on dealing with complexity and understandability issue in process mining. Besides research, he holds a wide range of experience in programming Microsoft.NET platform. He has also leaded the Web Team of IEEE Conference on Computers, Electronic and Electrical Engineering (ICE Cube), from 2016 to 2018.

MAIZATUL AKMAR ISMAIL (Member, IEEE) received the bachelor's degree from the University of Malaya, Malaysia, the master's degree from the University of Putra Malaysia, and the Ph.D. degree from the University of Malaya. She has more than 20 years of teaching experience since she started her career as a Lecturer at the University of Malaya, where she is currently an Associate Professor with the Department of Information Systems, Faculty of Computer Science and Information Technology. She was involved in various researches, leading to the publication of several academic papers in the areas of information systems specifically on educational technology, recommender systems, and data mining. She has been actively publishing more than 70 conference papers at renowned local and international conferences. A number of her works were also published in reputable international journals. She has participated in many competitions and exhibitions to promote her research works. She has been appointed as competition judges for several innovation competitions. To date, she has successfully supervised ten Ph.D. and 23 master's students to completion. She hopes to extend her research beyond information systems in her quest to elevate the quality of teaching and learning.

SURAYA HAMID (Member, IEEE) received the B.I.T. degree in industrial computing and the M.I.T. degree in computer science from The National University of Malaysia, in 1998 and 2000, respectively, and the Ph.D. degree from the Department of Computing and Information Systems, The University of Melbourne, in 2013. She is currently an Associate Professor with the Department of Information Systems, Faculty of Computer Science and Information Technology, University of Malaya, Malaysia. She was involved in various research, leading to the publication of several academic articles in the areas of information systems specifically on social informatics, educational technology, information services, e-learning, and big data initiative related.

MOHAMMAD HAIRUL NIZAM MD NASIR (Member, IEEE) received the Master of Computer Science degree from the University of Malaya, Malaysia, in 2005, and the Ph.D. degree in computer science from Universiti Teknologi, Malaysia, in 2014. He is currently serving as a Senior Lecturer at the Department of Software Engineering, Faculty of Computer Science and Information Technology, University of Malaya. He has published more than 70 scientific publications. His research interests include software process, software process improvement, project management, radio frequency identification, object-oriented design and development, accessibility, and computing. He is a member of the IEEE Computer Society. He has won several medals at the national and international exhibitions. As an active researcher, he has run several research projects and produced significant research output.