



HAL
open science

Stable Heuristic Miner 2: Evaluating the Statistical Stability in Event Logs to Discover Business Processes

Sina Namaki Araghi, Franck Fontanili, Elyes Lamine, Frederick Benaben

► **To cite this version:**

Sina Namaki Araghi, Franck Fontanili, Elyes Lamine, Frederick Benaben. Stable Heuristic Miner 2: Evaluating the Statistical Stability in Event Logs to Discover Business Processes. *Human-Centric Intelligent Systems*, 2024, 4, pp.256-277. 10.1007/s44230-024-00064-4 . hal-04511067

HAL Id: hal-04511067

<https://imt-mines-albi.hal.science/hal-04511067v1>

Submitted on 19 Mar 2024

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons Attribution 4.0 International License



Stable Heuristic Miner 2: Evaluating the Statistical Stability in Event Logs to Discover Business Processes

Sina Namaki Araghi¹ · Franck Fontanili² · Elyes Lamine² · Frederick Benaben²

Received: 15 September 2023 / Accepted: 6 February 2024
© The Author(s) 2024

Abstract

Automated process discovery as one of the paradigms of process mining has attracted both industries and academic researchers. These methods offer visibility and comprehension out of complex and unstructured event logs. Over the past decade, the classic heuristic miner and applied heuristic-based process discovery algorithms showed promising results in revealing the hidden process patterns in information systems. One of the challenges related to such algorithms is the arbitrary selection of recorded behaviors in an event log. The offered filtering thresholds are manually adjustable, which could lead to the extraction of a non-optimal process model. This is also visible in commercial process mining solutions. Recently, the first version of the stable heuristic miner algorithm targeted this issue by evaluating the statistical stability of an event log. However, the previous version was limited to evaluating only activities' behaviors. In this article, we'll be evaluating the statistical stability of both activities and edges of a graph, which could be discovered from an event log. As a contribution, the *stable heuristic miner 2* is introduced. Consequently, the definition of the descriptive reference process model has improved. The novel algorithm is evaluated by using two real-world event logs. These event logs are the familiar Sepsis data set and the urology department patients' pathways event log, which is recorded by monitoring the interpreted location data of patients on hospital premises and is shared with the scientific community in this article.

Keywords Process mining · Statistical stability phenomenon · Automatic process discovery · Heuristic miner

Abbreviations

<i>LCL:</i>	Lower control limit
<i>CL:</i>	Central line
<i>UCL:</i>	Upper control limit
<i>MR:</i>	Moving range
<i>LCL.edge:</i>	Lower control limit for edges
<i>CL.edge:</i>	Central line for edges

<i>UCL.edge:</i>	Upper control limit for edges
<i>Stable heuristic miner V2:</i>	Stable heuristic miner version 2

1 Introduction

Process mining is leading the process-oriented data science research paradigm that is predominantly known for extracting insights from event logs captured by information systems. It aims at analyzing and visualizing event logs to reveal process patterns. Process mining is identified by three main activities: automatic process discovery, conformance checking, and enhancement.

Over the past decade, more than 80 papers addressed the automatic discovery research issues by proposing new and variations of existing algorithms [1]. Often, these methods tend to extract models that differ in their approach to consider the trade-off among precision, fitness, and complexity criteria. Generally, the decision to respect which quality dimension criteria is arbitrarily made. Hence, it is difficult

✉ Sina Namaki Araghi
sina.namakiaraghi@uttop.fr

Franck Fontanili
franck.fontanili@mines-albi.fr

Elyes Lamine
elyes.lamine@mines-albi.fr

Frederick Benaben
frederick.benaben@mines-albi.fr

¹ Production Engineering Laboratory of École Nationale d'Ingénieurs de Tarbes (ENIT), Université de Technologie Tarbes (UTTOP), 65016 Tarbes, France

² Industrial Engineering Center of IMT Mines Albi, 81000 Albi, France

to detect a reference model using conventional process discovery methods.

For instance, the heuristic-based process discovery methods captivated the attention of researchers in the discipline of process mining. They have been identified as a proper solution to discover less structured process models. This is due to their ability to handle noises in an event log [2, 3]. The work in [4] was one of the pioneers in developing heuristic-based process discovery algorithms, and the developed algorithm was named the *heuristic miner*. It can be inferred that the main objective of such an algorithm was to give flexibility to users so that they can detect the target model from an event log. This has been observed in improved versions of this algorithm [5].

The classic heuristic-based methods are developed mainly in five general steps which is (i) *identifying the footprint matrix*, (ii) *calculating the dependency measure*, (iii) *devising the graph*, (iv) *discovering the splits and joins*, and (v)

adjusting the loops. In this article, we will elaborate on the second step of this algorithm. This decision is founded upon one of the scientific gaps that we have identified, and we will elaborate on it in the subsequent section.

1.1 The Scientific Gap and Its Risks

The actions to extract the footprint matrix lead to detecting the direct relations among activities. The advantage of the classic heuristic miner algorithm is within its second step where the algorithm calculates the dependency measure among activities. By defining this metric, the algorithm gives the mentioned *flexibility* advantage to select a target model [5].

This so-called flexibility is offered to users by providing manually adjustable thresholds, which is a common approach of most process discovery algorithms. These thresholds allow modification of the graph by *adding* or

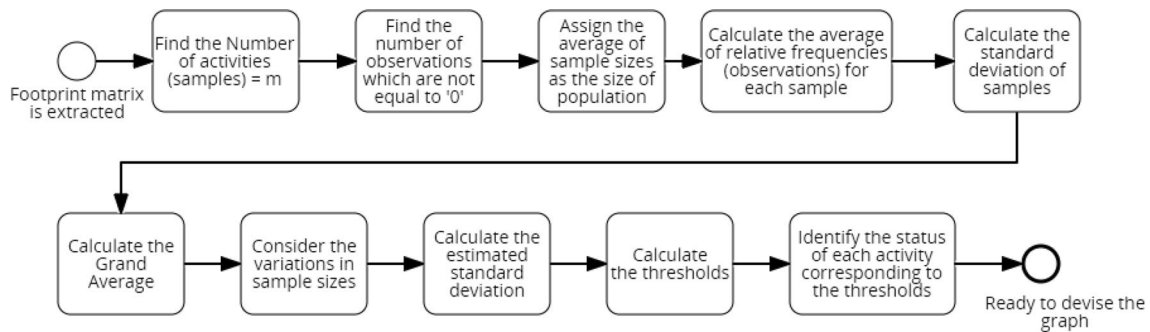


Fig. 1 Illustration of the steps of the first version of the stable heuristic miner to evaluate the statistical stability for activities [10, 12]

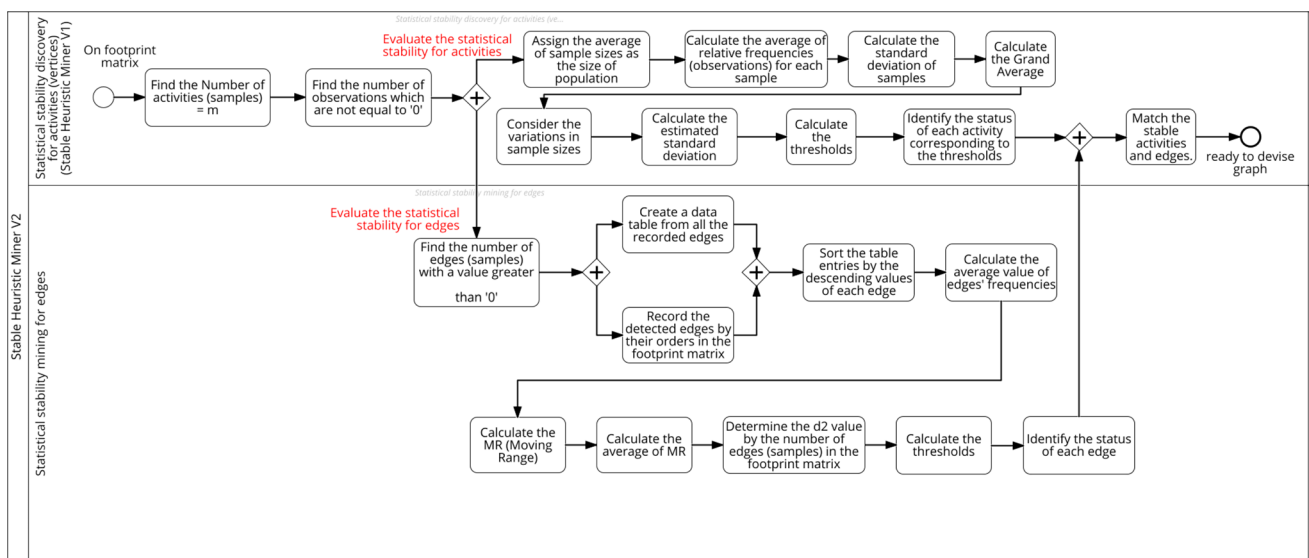


Fig. 2 Stable heuristic miner 2 algorithm: the sequence of actions that lead to the output of the algorithm

Table 1 Presenting the value of *observations* and the *average Moving Range (MR)*

Order	FirstActivity	SecondActivity	× (Edge value)	MR
1	a	c	34	0
2	a	b	26	8
3	b	c	46	20
4	b	i	5	41
5	b	k	4	1
6	b	f	1	3
7	c	d	52	51
8	c	b	31	21
9	c	i	10	21
10	c	f	6	4
11	c	j	6	0
12	c	e	5	1
13	c	h	4	1
14	d	l	23	19
15	d	g	17	6
16	d	e	14	3
17	e	l	20	6
18	e	c	5	15
19	f	c	15	10
20	f	d	2	13
21	g	l	11	9
22	g	h	5	6
23	g	e	1	4
24	h	e	5	4
25	h	c	4	1
26	i	c	9	5
27	i	f	6	3
28	j	l	6	0
29	k	e	4	2
30	l	m	60	56
$\bar{x} = 14.566$			$\bar{MR} = 11.133$	

removing activities (vertices) and connections (edges) among them. This modification is mainly carried out arbitrarily, and it's highly dependent on the experience and knowledge of the user. This is incompatible in case the user wants to understand how a process normally is being executed in reality, and the detection of drifts is not possible. This shortcoming was evoked in other works as well [6]. Additionally, in data-driven simulation, one of the main challenges is to extract the reference behavior recorded in an event log to eventually identify a simulation model.

Moreover, the extracted results are likely to illustrate a non-optimal graph. Increasing the value of these thresholds could also lead to detecting uncommon behaviors, more entropy, and impractically enlarging the size of the graph.

This gap could be observed in other studies too [6–9]. For instance, researchers in [9] mentioned that the *quality of the discovered model depends on trial-and-error and can be time-consuming to detect a target model*. Therefore, considering the second step of these algorithms, the research question for heuristic-based process discovery algorithms is “*how to calculate the optimal values for thresholds?*”

The primary version of the *stable heuristic miner* algorithm was proposed to address this research question [10–12]. To do so, the stable heuristic miner proposed to evaluate the statistical stability in an event log. As a result, a process model is discovered that represents the *descriptive reference process model*. This algorithm was developed to detect a reference behavior of patients with the objective of diagnosing deviations and drifts from it [11]. This was seen as an important requirement in the analysis of patients' pathways. This was due to the fact that the healthcare processes are highly complex, and users need a reference model for diagnosis actions, which is difficult to obtain.

However, the previous version of the stable heuristic miner addressed the evaluation of statistical stability by focusing only on activities (vertices). As a result, the discovered process model could not effectively consider the statistical stability of the whole behavior registered in an event log (i.e., activities and relationships among them). This was one of the concerns highlighted in [10, 12]. In this article, we aim to address this issue.

1.2 Hypothesis and Contributions

In complex systems with emergent properties like a hospital or environments where humans are running the majority of tasks, it's a challenge to capture a model that illustrates the reference behavior of a statistical population. To address this issue, we define *statistical stability in process discovery*.

We establish that *Statistical stability in process discovery* is manifested through a *meta-analysis that evaluates the consistency of samples' behaviors against the recorded information in an event log, which we consider as a statistical population*.

Each sample is identified by an activity or a connection between two activities. By evaluating the statistical stability, one could obtain a snapshot that can be used as the main and reference behavior of the population. This evaluation—statistical stability—is carried out by *assessing the frequency of mass events, and also by taking the stability of averages, variances, and the standard deviations of the samples into consideration* [13, 14]. Accordingly, the contribution of this article is:

- The new *stable heuristic miner 2* algorithm, which redefines the descriptive reference process model by assess-

ing the statistical stability of both activities and edges extracted from an event log.

- A real-world event log, which is used to assess the capability of the novel algorithm, which is accessible in [15].

1.3 Article’s Structure

The remainder of the article is organized as follows: Sect. 2 collects the main state of the art of the related works addressing the heuristic-based mining methods for process discovery. Section 3 describes the developed algorithm and its definitions. Additionally, it provides a running example for a better illustration of the introduced method. Section 4 reports two experiments to validate the applicability of the new algorithm, and it provides a comparison with previous methods. These experiments are based on two real-world event logs. Section 5 summarizes the main conclusions of this work, its limitations, and its potential future works.

2 Background

2.1 A Review of Heuristic-Based Process Discovery Methods

As mentioned in Sect. 1.1, calculating the dependency measure is a principal step targeted by several heuristic-based process discovery algorithms. Optimizing the output of this step could lead to the extraction of an optimal dependency graph, or as mentioned here the descriptive reference process model.

Previously in [4, 5, 16], the dependency graph was defined as:

Dependency Graph

$$= \{(a, b) \mid (a \in E \wedge b \in a\square) \vee (b \in E \wedge a \in \square b)\} \quad (1)$$

This definition expresses ‘E’ as a limited set of activities. Each event could represent an activity. ‘□b’ stands for the activities that come before ‘b’. ‘a□’ denotes the activities that come after ‘a’. Accordingly, a dependency relation is represented by (a, b) expressing the *input – output* and sequence of activities.

Therefore, the dependency measure [5] could be expressed by equation 2:

$$Dependency\ Measure : a \Rightarrow_w b = \frac{|a >_w b| - |b >_w a|}{|a >_w b| + |b >_w a| + 1} \quad (2)$$

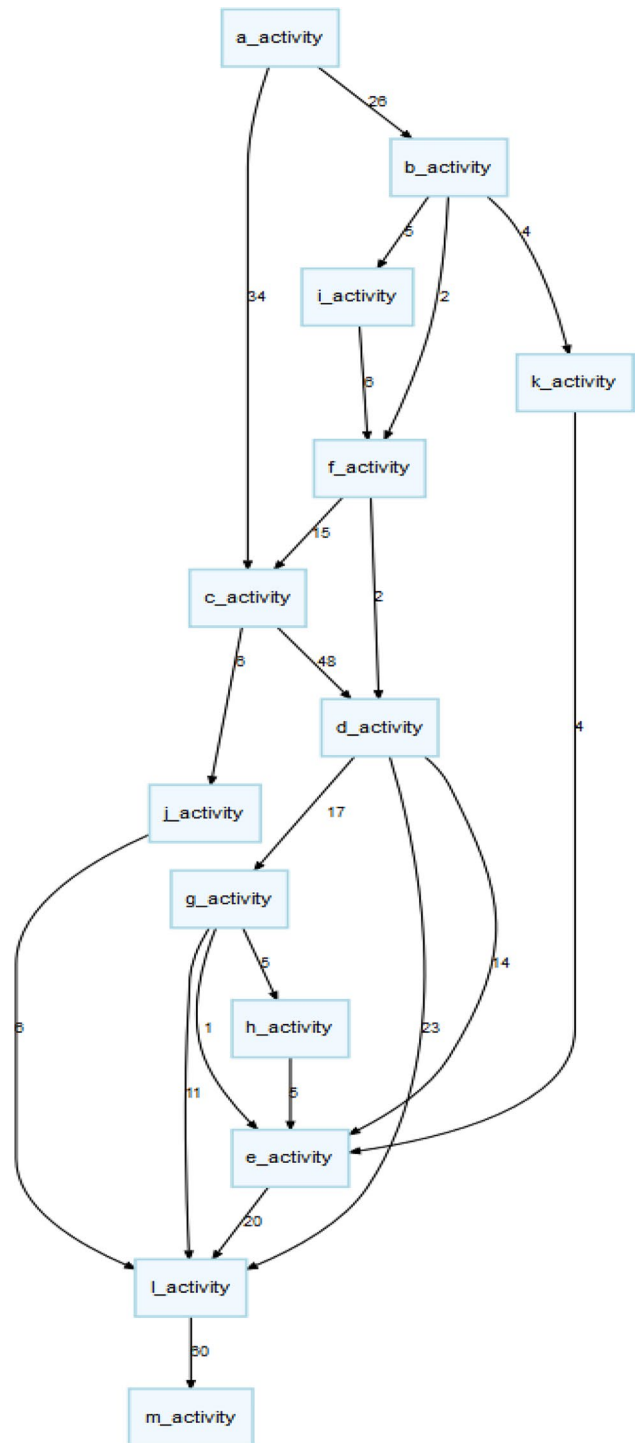


Fig. 3 The process model extracted by the classic heuristic miner approach with a manual threshold set at 20%. Accordingly, the model presents activities that have a dependency measure higher than 20%. Only 67% of the recorded behaviors respect this threshold

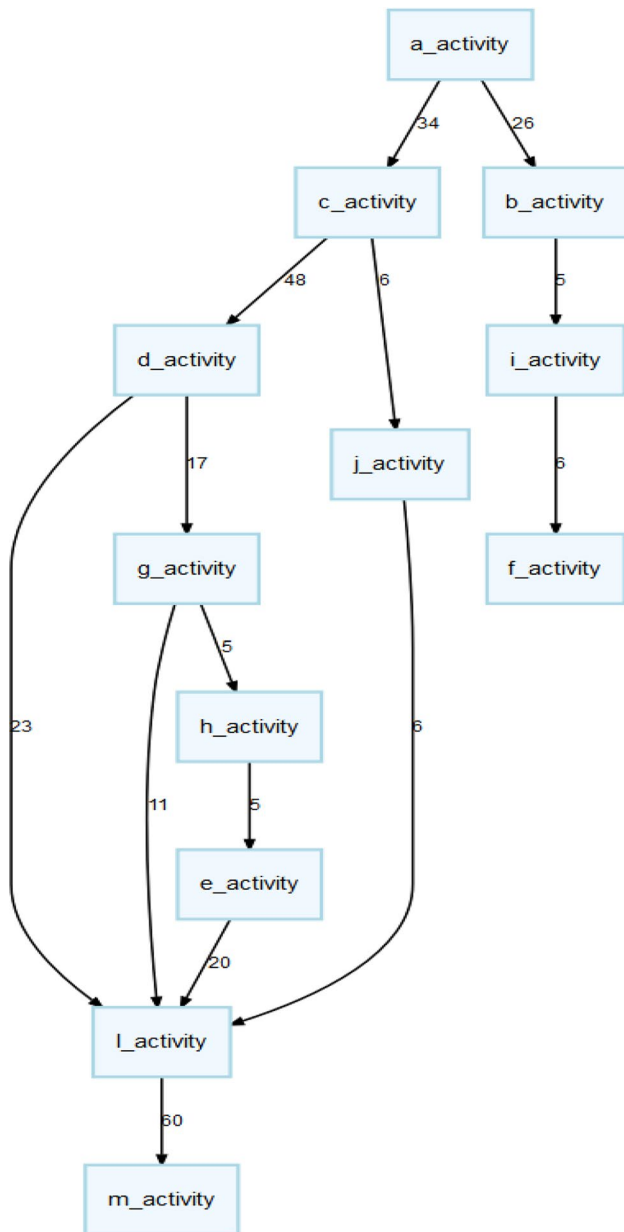


Fig. 4 The process model extracted by the classic heuristic miner approach with a manual threshold set at 80%. Accordingly, the model presents activities that have a dependency measure higher than 80%. Only 45% of the recorded behaviors respect this threshold

Here, ‘ w ’ stands for the event log with ‘ n ’ number of activities. $|a >_w b|$ denotes the number of times activity ‘ a ’ is followed by ‘ b ’. An increase or decrease in the value of these thresholds by the user will lead to a set of vertices and edges. This selection is arbitrary and doesn’t guarantee an optimal graph.

The primary heuristic miner algorithm [4] discovered the dependency graph based on the *minimum thresholds* for evaluating the dependency among activities.

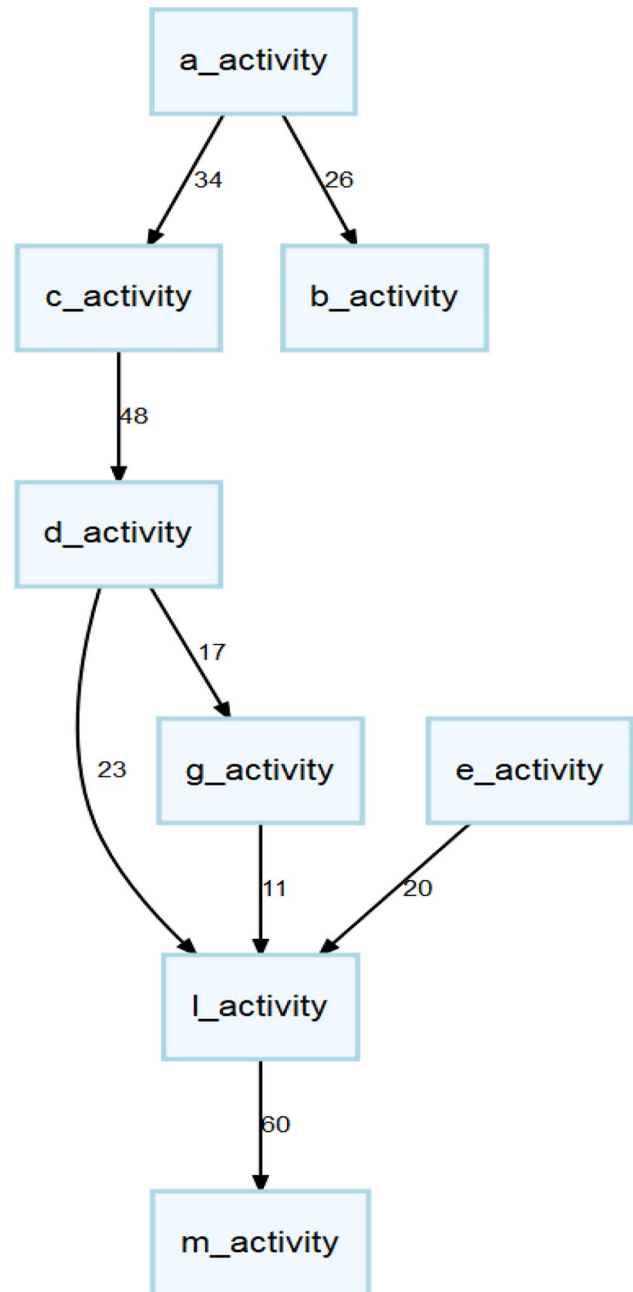
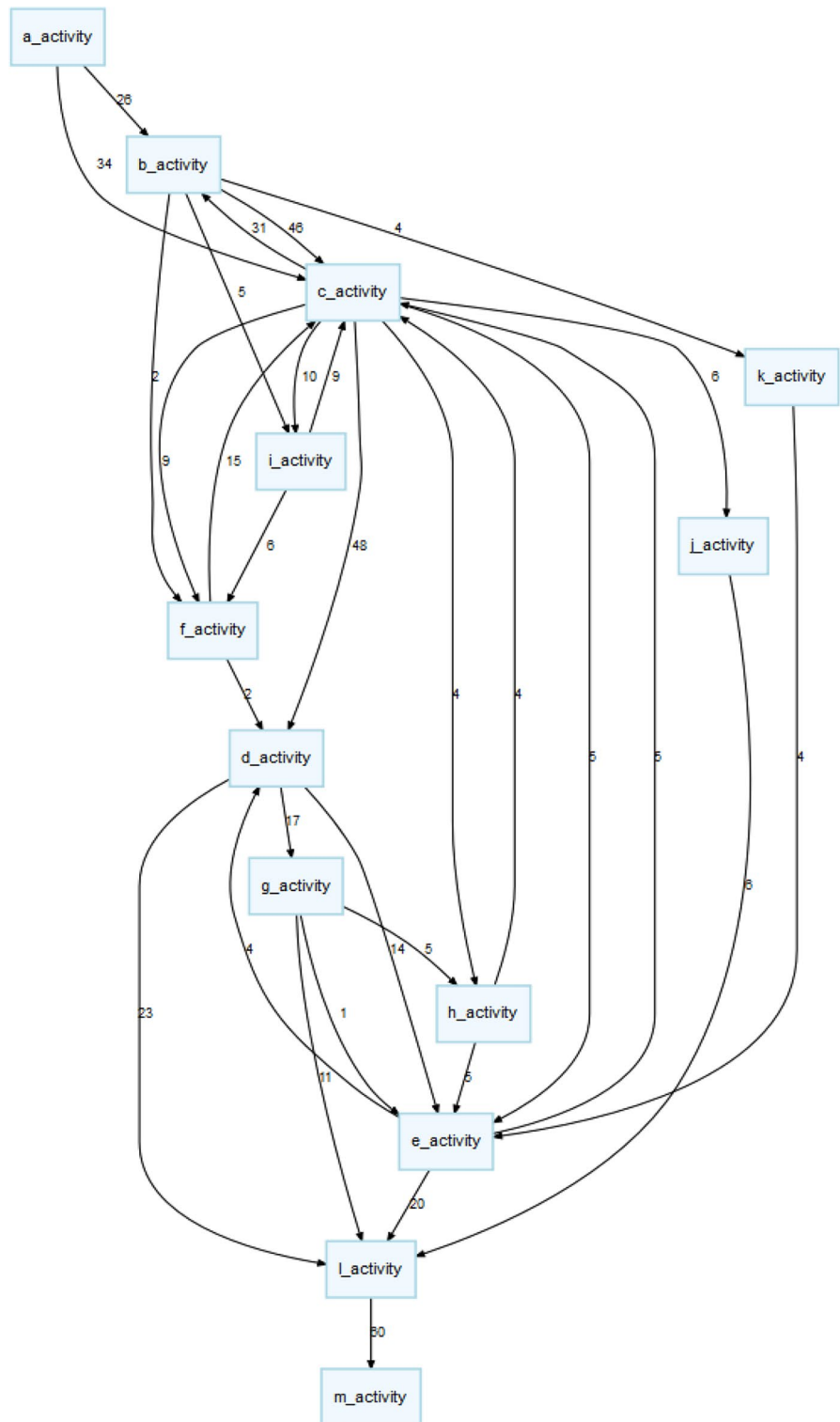


Fig. 5 The process model extracted by the classic heuristic miner approach with a manual threshold set at 90%. Accordingly, the model presents activities that have a dependency measure higher than 90%. Only 25% of the recorded behaviors respect this threshold

Authors in [17] presented another version of the heuristic miner by focusing on the mentioned scientific gap. They proposed a new approach to calculate the dependency measures. One of the advantages of this version was the ability to extract dependency graphs from event streams.

Heuristic miner ++ was introduced in 2015 [18]. This version also tried to modify the approach in calculating the

Fig. 6 The process model extracted by the classic heuristic miner approach with manual configuration of thresholds. The value for the threshold is set at 0, therefore, the model shows all the registered behaviors



dependency measures by considering the time intervals in an event log.

Another version of the heuristic miner was proposed as the “flexible heuristic miner” in [5]. The output of this version was constructed similarly to the classic heuristic miner,

however, it improved the results by considering long-distance dependencies as well.

The Fodina algorithm [19] is another improved version of the heuristic miner, which is capable of detecting duplicate tasks and providing more flexibility for the user to extract a

Table 2 Presents the evaluation of process discovery results of the running example considering the quality criteria mentioned in the literature

Discovery approach	Precision	Fitness	Complexity in size	Complexity in behavior	Logic of substracting information
Classic heuristic miner (Fig. 6)	0.1	1	1	1	All behaviors of the log are selected
Classic heuristic miner (Fig. 3)	0.4	1	1	0.8	Dependency measure over 0.2
Classic heuristic miner (Fig. 4)	0.4	0.9	0.9	0.5	Dependency measure over 0.8
Classic heuristic miner (Fig. 5)	0.3	1	0.6	0.3	Dependency measure over 0.9
Stable heuristic miner 1st version (Fig. 7)	0.1	1	0.8	0.9	Evaluating statistical stability by considering each activity in an event log as a sample of a statistical population which is the event log
Stable heuristic miner 2nd version (Fig. 8)	0.66	0.67	0.7	0.51	Evaluating statistical stability by considering each activity AND each edge's behaviors in an event log as a sample of a statistical population which is the event log

dependency graph. Authors in [8] mentioned that this algorithm could lead to disconnected graphs, which could be seen as irrelevant.

These presented works aimed at improving the heuristic miner functionalities. However, their aim was narrowly different from the focus of the stable heuristic miner 2. In this new algorithm, we focus on the extraction of one target model from the data for specific analytical purposes such as diagnosis and concept drift detection.

Several researchers in [8, 20–24] worked on mathematical programming applications to address the research question of “how to discover an optimal process”. For instance, authors in [21] considered integer linear programming. They try to optimize their functions according to an interesting constraint, which is assuring the modeled activity is on a path starting from the initial activity to the end activity. The Proximity miner algorithm [22] is an interesting example of these works, which integrates the domain knowledge in the discovery task of the dependency graph. Consequently, the scientific question appears here again that the mining procedure is dependent on the expert’s knowledge.

We identified two promising methods with an approximately similar objective to ours. The first one is the Inductive Miner algorithm [25], which is one of the most applied process discovery algorithms. This method aims to detect the most significant splits in an event log and determine associated operators to characterize each split. This method results in block-structured process models, which are visually appealing. However, such a model could generate a flower-structure process model, leading to low fitness and generating behaviors that were not seen in the event log. Moreover, this method identifies various process structures and suggests thresholds to detect linear models. Considering its approach, it did not match our plan to propose a reference model based on mathematical logic. It has been seen that both Inductive Miner and classic Heuristic Miner have been ineffective in detecting the relevant/reference process model

[6]. In [6], authors compared the effectiveness of classic heuristic miner and inductive miner while using the sepsis data set—used in this paper as well—and highlighted the need to develop methods for advancing towards extracting a reference process model.

The other method is the Split Miner algorithm [26]. The objective of this promising method is to extract *simple process models with low branching complexity* while keeping a high fitness value. This approach does not predominately lead to a reference model, since it does not consider the statistical distribution of the events nor the nature of the observed system. The application of this method to our study could lead to the extraction of a simple model, but not necessarily the descriptive reference model.

2.2 The Previous Version of the Stable Heuristic Miner Algorithm

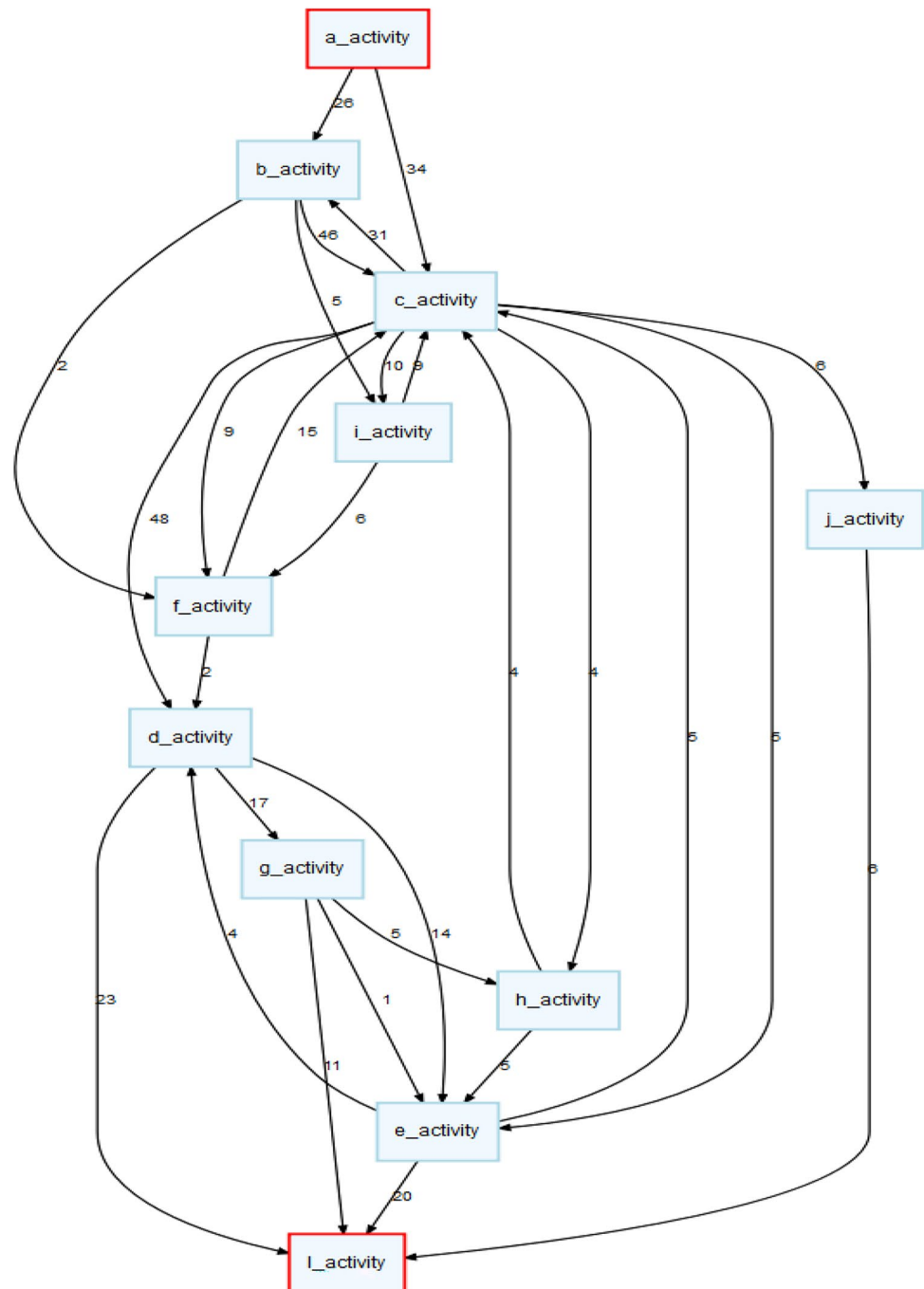
As mentioned earlier, in the first version of the stable heuristic miner [10, 12], we aimed to find the presence frequency of all activities in an event log. Then, we obtained three main thresholds to see how important the activity is according to the statistical stability phenomenon.

The reason statistical stability was found to be an appropriate approach to evaluate and discover a process model from an event log was based on *the nature of the studied system*.

According to the hypothesis of statistical stability [13, 14], to understand the behavior of a complex system, *it is required to evaluate the behavior of each member of the system based on its impact on the whole system*. To our humble knowledge, this is a missing notion in the core of most process discovery algorithms.

The statistical stability evaluation could be driven by the Shewhart control charts approach [10, 27]. These control charts include a *center line (CL)* that represents the average value of a measured characteristic, relevant to the in-control

Fig. 7 The process model extracted by the first version of the stable heuristic miner. Red activities have shown a high level of variation in their behavior according to the algorithm



state. The *Upper Control Limit (UCL)*, and *Lower Control Limit (LCL)* are calculated by considering the standard deviations and averages of the samples. These two limits—UCL & LCL—are the borders of a statistically stable state. As long as the analyzed data points fall between these two thresholds,

the outcomes of the process are in control. If a data point falls outside these limits, it will be considered as a variation of the process outcomes, and the process will no longer be considered stable.

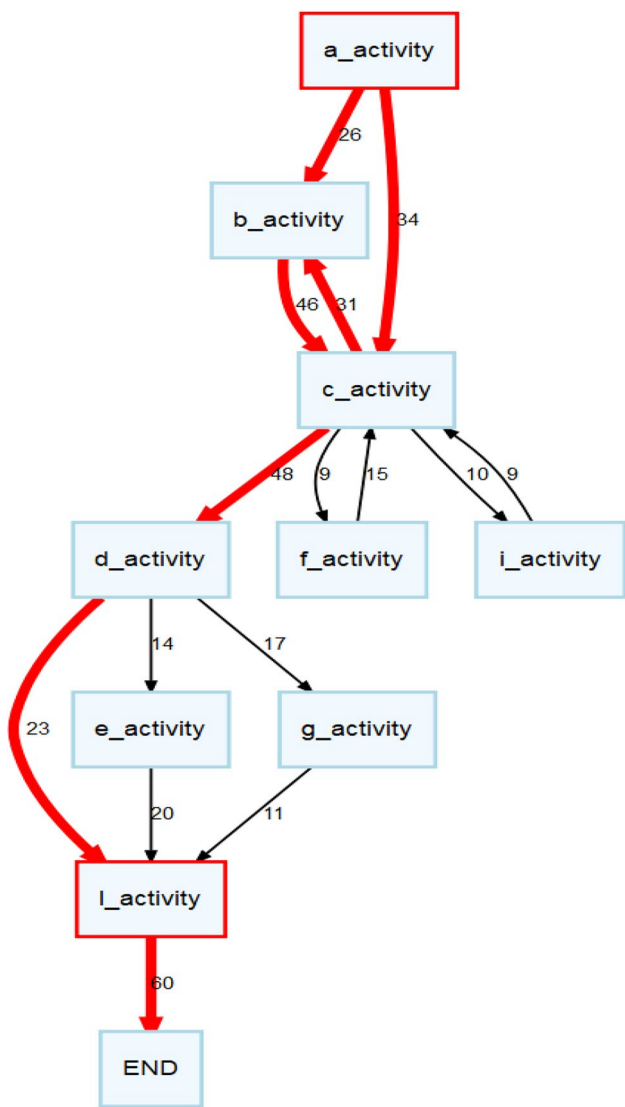


Fig. 8 The model extracted by the new version of the stable heuristic miner. Red edges and activities have shown a high level of variations in their behavior according to the algorithm

As a result, the previous version of the stable heuristic miner algorithm proposed to replace the second step of the classic heuristic miner with ten new steps.

These steps were defined as the sequence of actions to identify the *stable activities* and *activities that are imposing instability into the process*. These steps are shown in Fig. 1. By identifying these two types of activities, a new definition was proposed in equation 3. This definition represents a *descriptive reference process model*.

According to equation 3, in order to acquire the descriptive reference process model, we need to detect a state from

the event log in which for each activity (\mathcal{A}), a vector as s represents the frequency of relations with other activities. ‘ s ’ vector is defined as a sample of the population (footprint matrix). Additionally, for each ‘ s ’ sample, there exists a \bar{x}_s , which is the average of direct relations frequencies. Therefore, the corresponding activity to the sample (s) will be shown in the descriptive reference process model (\mathcal{P}) if the average of its direct relations frequencies is between the two thresholds; ‘ UCL ’ and ‘ LCL ’. Consequently, such a sample (activity) is considered a stable behavior. If the average value is greater than the UCL value, then, it’s considered a hot zone that imposes instability into the process.

$$\begin{aligned}
 & [\forall \mathcal{A} \exists s] \wedge [\forall s \subseteq \mathcal{S} \exists \bar{x}_s] \\
 & \therefore \\
 & (\mathcal{A} \in \mathcal{P}) \rightarrow [LCL < \bar{x}_s < UCL] \cup [UCL \leq \bar{x}_s]
 \end{aligned}
 \tag{3}$$

This definition—equation 3—evaluated the statistical stability criteria to detect solely the stable, and unstable activities behavior. On the other hand, each edge can have an impact on the nature of a graph, but this is ignored here.

Therefore, we propose an improvement in the following. The second version of this algorithm not only considers the behavior of activities in a dependency graph but also the edges that can be discovered from an event log. The following section presents the new modification to the stable heuristic miner algorithm.

3 Theory and Methods

This novel version will consider both activities (vertices) and relationships among them (edges) as samples of a larger statistical population, which is the event log.

3.1 Presenting the Steps of the Novel Algorithm

To better illustrate this new algorithm, we are going to present each step by using a running example. Figure 2 demonstrates the procedure of the second version of the algorithm.

To avoid redundancy, we will not focus on the steps related to the previous version of the algorithm, and we will present through a running example the definitions regarding the new improvements.

3.1.1 The Running Example

‘ L ’ represents a series of traces that are supposedly extracted from an event log.

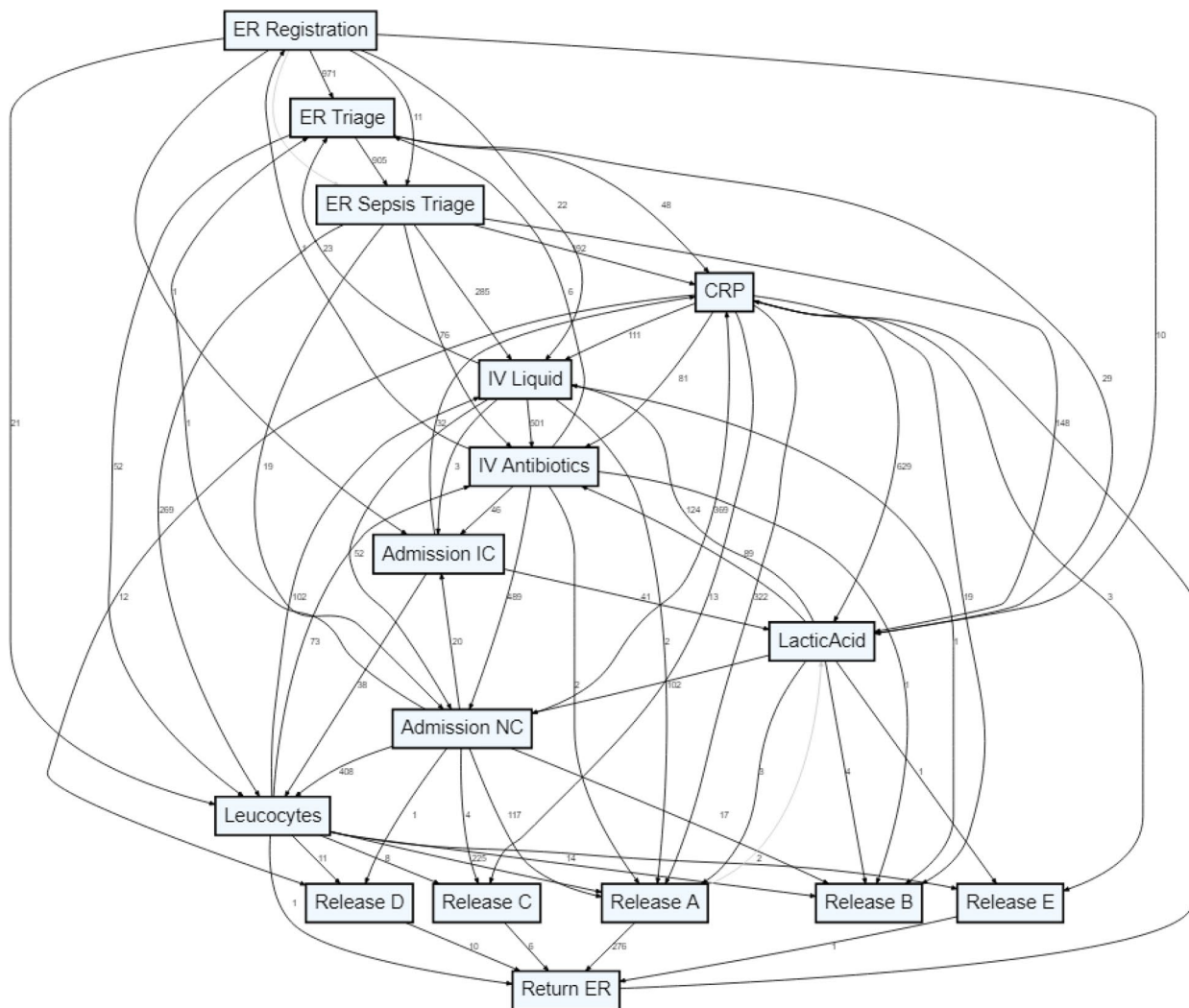


Fig. 9 Visualizing the sepsis process model by using the classic heuristic miner and applying a dependency threshold of 15%. The selection of the threshold value is random due to the approach of the algorithm

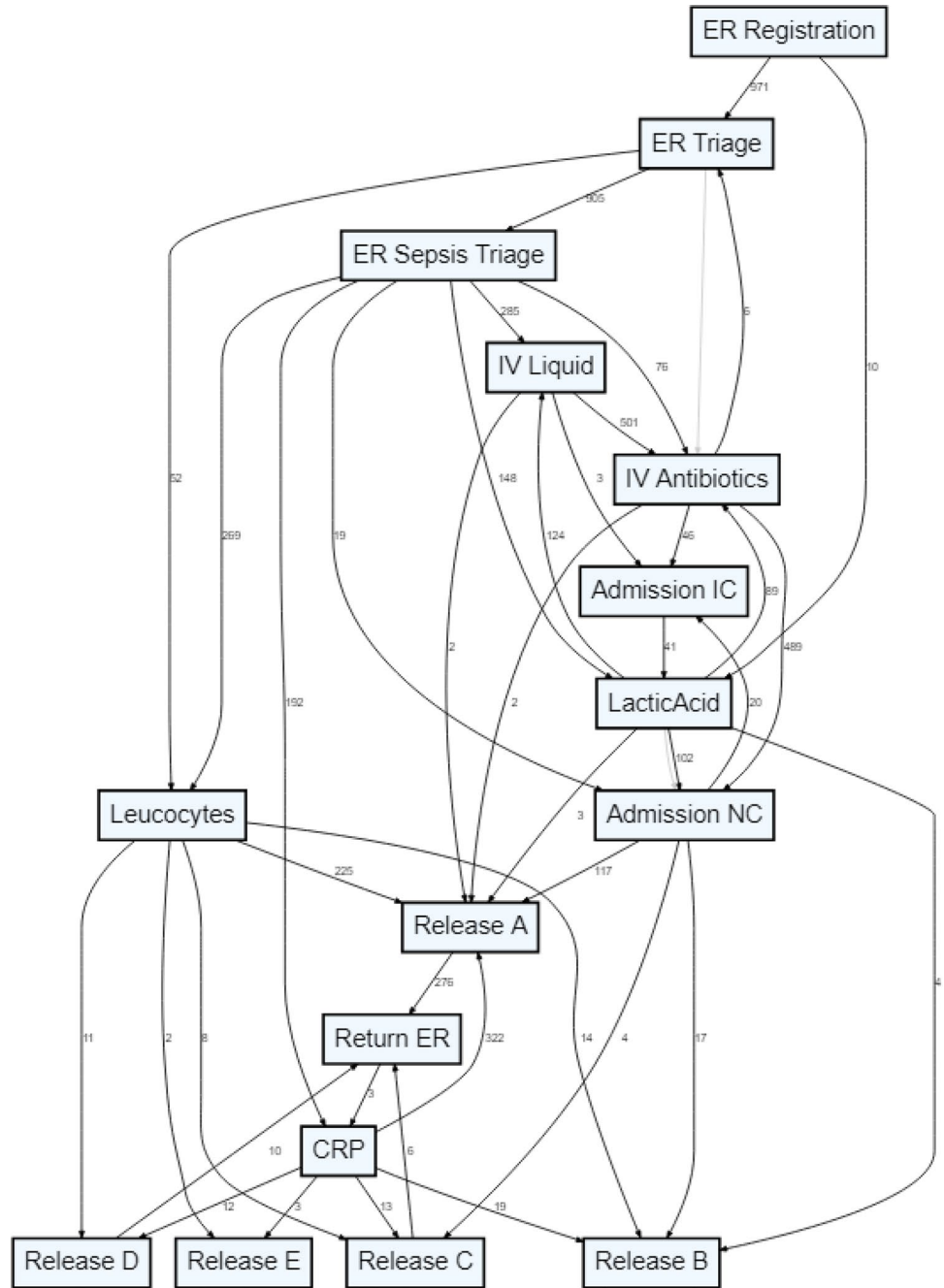
$$L = [\langle a, b, c, d, e, l, m \rangle^{12}, \langle a, b, f, d, e, l, m \rangle^2, \langle a, b, c, d, g, e, l, m \rangle^1, \langle a, b, c, d, g, h, e, l, m \rangle^5, \langle a, c, b, c, d, l, m \rangle^6, \langle a, c, f, c, d, l, m \rangle^3, \langle a, c, b, i, c, e, c, d, g, l, m \rangle^5, \langle a, c, b, c, f, c, d, l, m \rangle^6, \langle a, c, b, c, i, c, h, c, d, l, m \rangle^4, \langle a, c, b, c, i, f, c, d, g, l, m \rangle^6, \langle a, b, c, j, l, m \rangle^6, \langle a, c, b, k, e, d, l, m \rangle^4]$$

Inside L each group represents a trace. A trace consists of events corresponding to the activities. For instance, the first trace shows that 12 cases have followed the same sequence of activities. Figure 2 shows the new steps we are going to take to discover the descriptive reference

process model. We'll describe these steps in the following definitions.

After extracting the footprint matrix (shown below), we will consider the matrix as the *population* and each value as a *sample* (s_e).

Fig. 10 Visualizing the sepsis process model by using the classic heuristic miner and applying a dependency threshold of 50%. The selection of the threshold value is random due to the approach of the algorithm



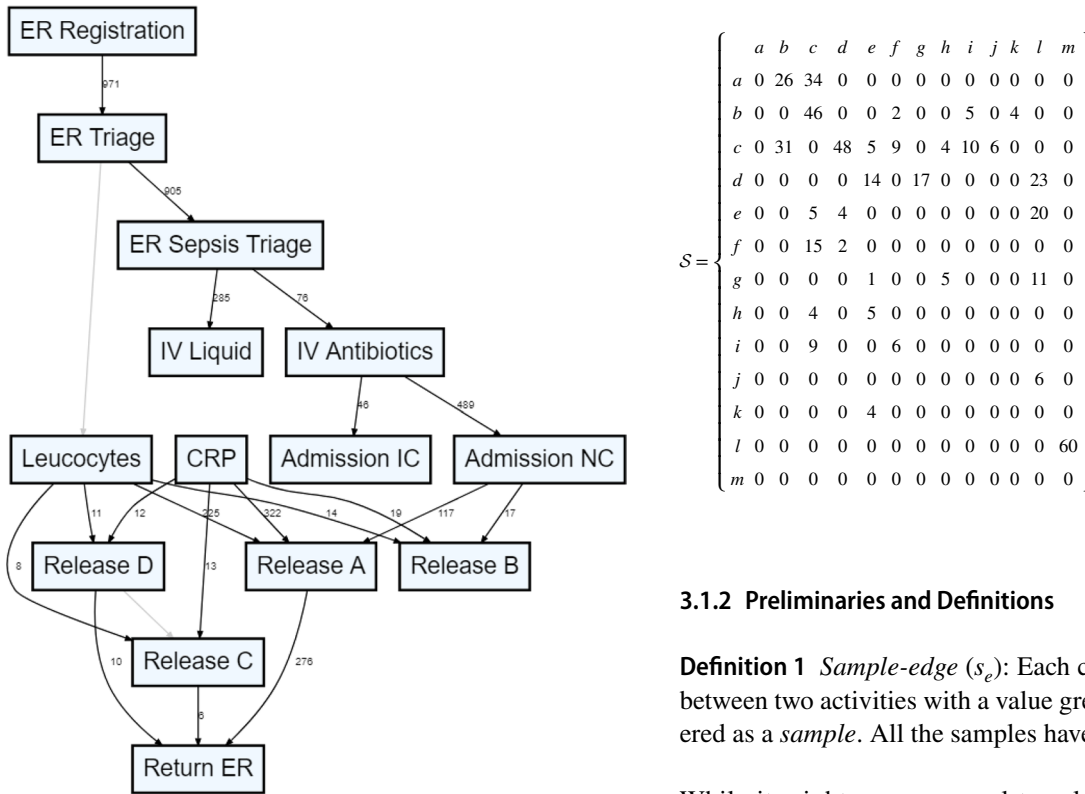


Fig. 11 Visualizing the sepsis process model by using the classic heuristic miner and applying a dependency threshold of 80%. The selection of the threshold value is random due to the approach of the algorithm

3.1.2 Preliminaries and Definitions

Definition 1 *Sample-edge* (s_e): Each connection or an edge between two activities with a value greater than 0 is considered as a *sample*. All the samples have a unique size of 1.

While it might seem unusual to select samples with the size of 1, there are statistical constraints that require such selection [28]. Accordingly, in the example above, there are ‘30 + 1’ samples (edges). The ‘+1’ is an assumption for presenting the connection to the *end activity*.

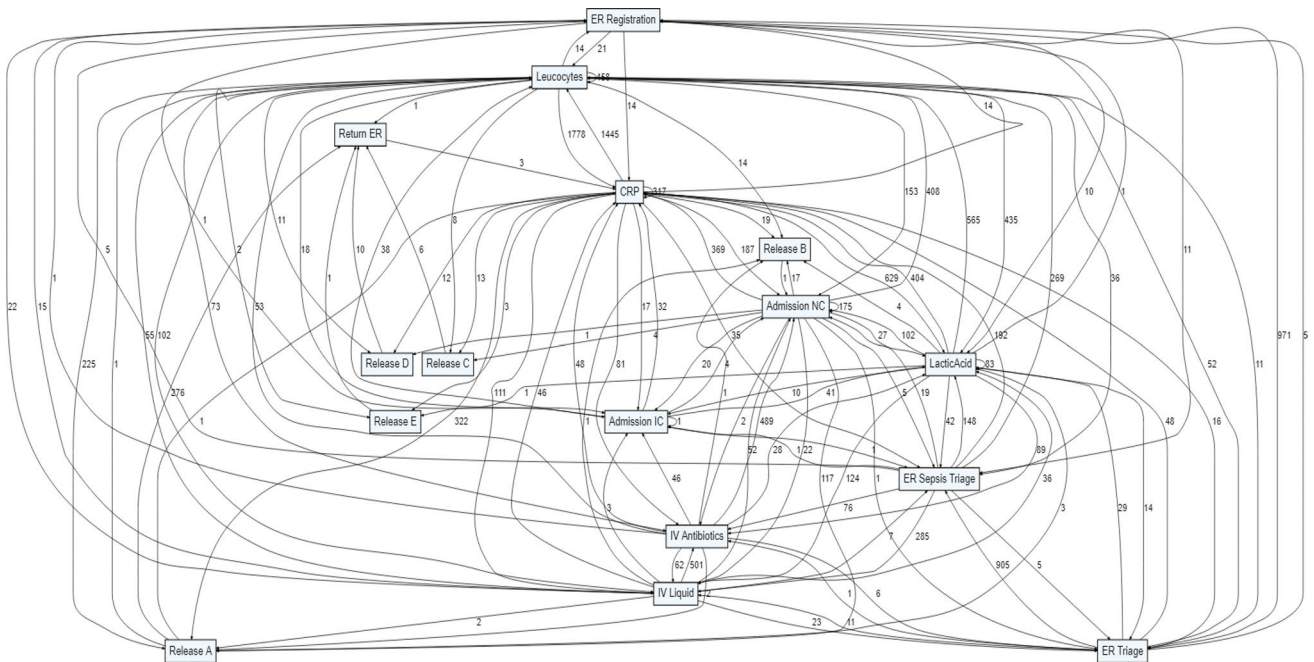


Fig. 12 Result of the stable heuristic miner v1 by using the Sepsis data set

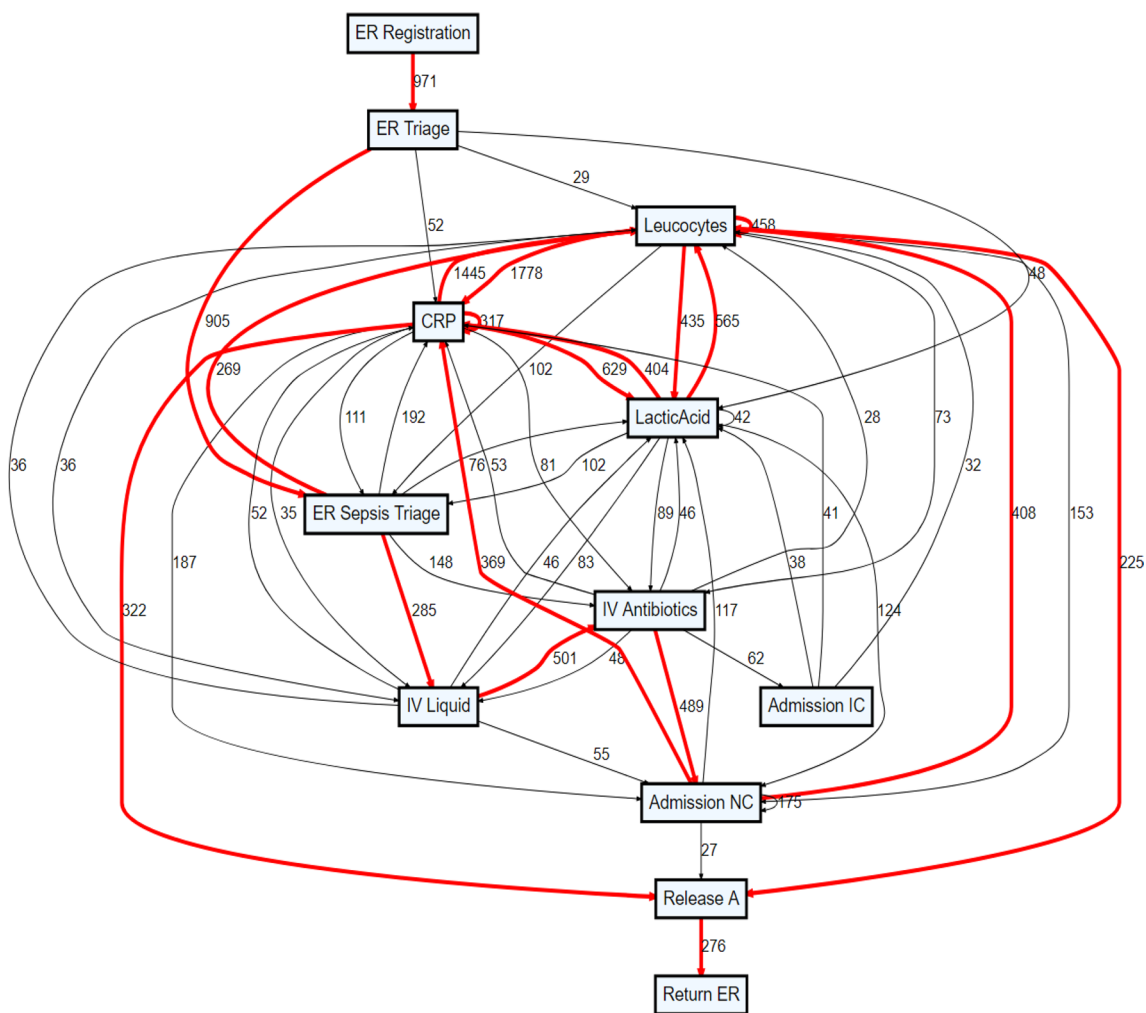


Fig. 13 Result of the new stable heuristic miner v2 algorithm by using the Sepsis data set

Definition 2 *Observation-edge* (x_{ij}): Each edge or sample within the population has a value. This value is called the *Observation* value.

As an example, in the previous footprint matrix, the value of $a \rightarrow b$ is equal to 26. Therefore, $a \rightarrow b$ is a sample of the population. The value of this sample is considered as an *observation*.

The value of each observation will be used to build upon the extraction of a statistically stable state. Needless, to mention, there are 31 edges in this example and accordingly ‘31’ observation values.

To understand the statistical behavior of each sample, we need to evaluate it by considering other samples’ behaviors. To do so, we define the *Moving range (MR)*.

Definition 3 *Moving Range (MR)*: *MR* represents the difference from one *observation*(x_{ij}) to another.

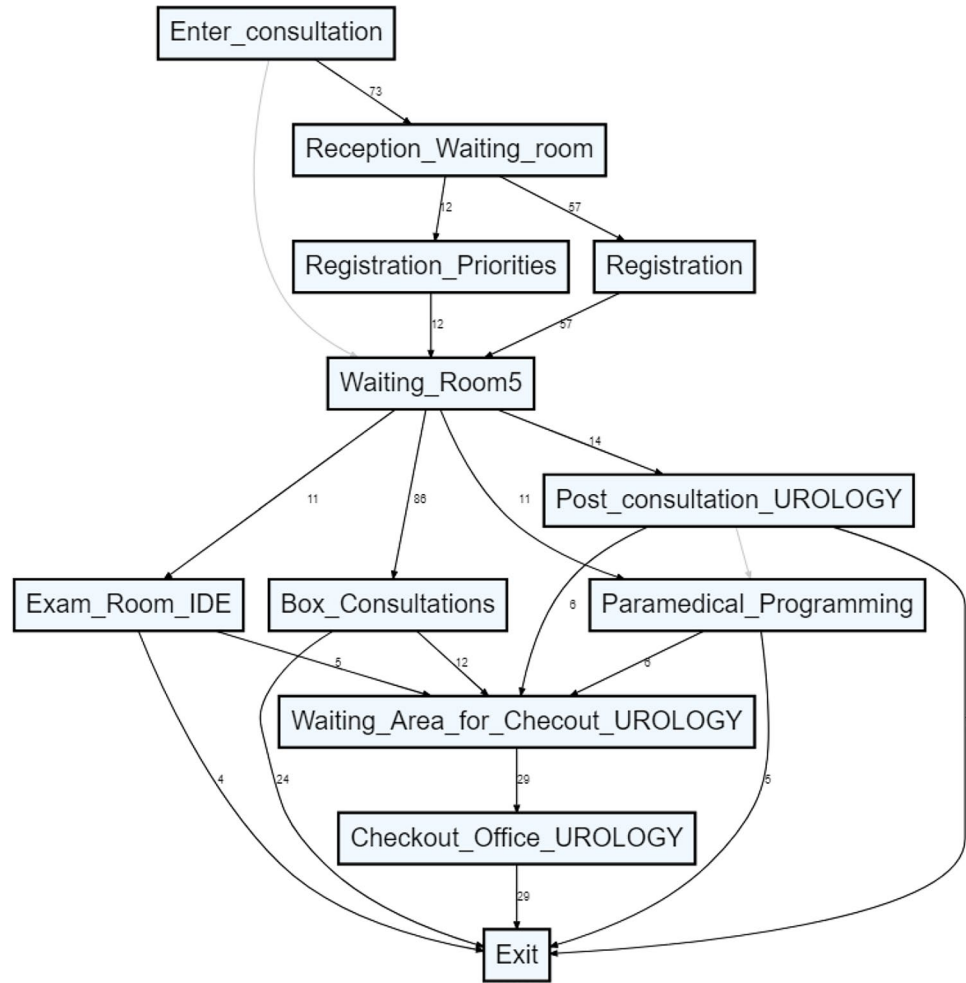
For instance, the $x_{ab} = 26$ and $x_{ac} = 34$. Therefore, the *MR* from x_{ab} to x_{ac} is equal to ‘8’. The order is based on the sequence in which events are recorded.

The value of *MR* helps to consider the variations in samples’ behaviors.

Definition 4 *Average behavior* (\bar{x}): As it can be expected, this value represents the average of all the samples(edges) values.

$$\bar{x} = \frac{x_1 + x_2 + x_3 + \dots + x_n}{n} \tag{4}$$

Fig. 14 Visualizing the urology process model by using the classic heuristic miner and applying a dependency threshold of 15%. The selection of the threshold value is random due to the approach of the algorithm



Definition 5 *Average of Moving Range (\bar{MR})*: While the value for each edge changes from one to another, the \bar{MR} gives a value to represent the average of variations.

$$\bar{MR} = \frac{MR_1 + MR_2 + MR_3 + \dots + MR_n}{n} \tag{5}$$

To calculate the value of \bar{MR} for the running example (L), at first, we sorted the observation values. Then, the average behavior \bar{x} was calculated. Finally, MR values or the changes in the values of edges will be calculated. This will lead to calculating the average of moving ranges (\bar{MR}). The result of these actions are shown in Table 1.

Accordingly, the average value of variations is $\bar{MR} = 10.51$. The average value for the recorded behavior of edges is $\bar{x} = 14.225$.

Definition 6 *Central Line-edges ($CL.edges$)*: This central threshold determines the most statistically stable edges (samples). If the value of an edge is closer to this line, It is more likely for this edge to be seen in future behaviors.

As shown in equation 6, the most stable behaviors are close to the average behavior.

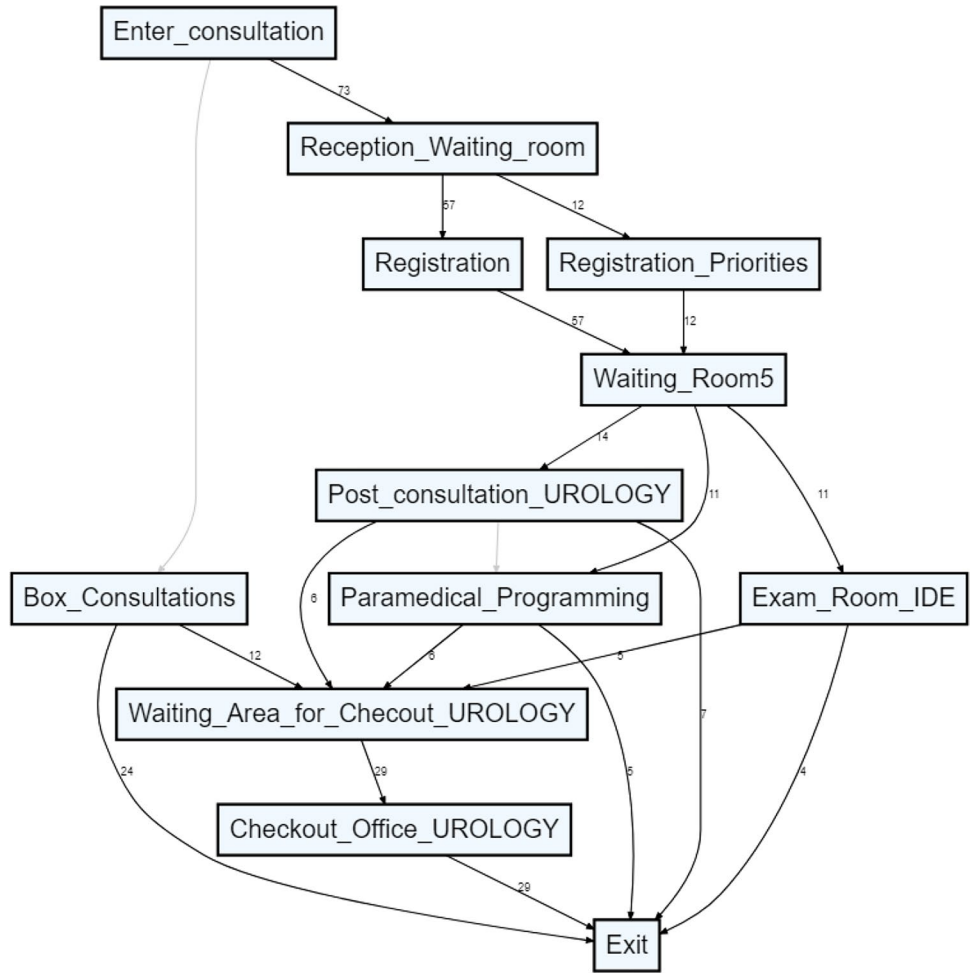
$$CL.edges = \bar{x} = \frac{x_1 + x_2 + x_3 + \dots + x_n}{n} \tag{6}$$

Definition 7 *Upper Control Limit-edges ($UCL.edges$)*: This threshold determines the limit to conceive edge behavior as stable. If an edge value passes this threshold, it will be considered as an edge with a high variation ratio in its behavior.

Equation 7 presents the mathematical model of UCL . A customary constant as ' d_2 ' is defined here. The value for ' d_2 ' is defined through a series of calculus operations and it has led to certain constant values. d_2 values are depending on the number of monitored samples in a population. Additionally, these values are presented in most statistics handbooks [28].

$$UCL.edges = \bar{x} + 3 \left(\frac{\bar{MR}}{d_2} \right) \tag{7}$$

Fig. 15 Visualizing the urology process model by using the classic heuristic miner and applying a dependency threshold of 50%. The selection of the threshold value is random due to the approach of the algorithm



Definition 8 Lower Control Limit-edges (*LCL.edges*): The *LCL* value ascertains the unstable behaviors. The edges with values less than *LCL* are not expressed within the discovered process model. This threshold helps to remove the so-called “dirt roads” from a process model automatically.

Equation 8 refers to the mathematical model for extracting *LCL* value.

$$LCL.edges = \bar{x} - 3 \left(\frac{\bar{MR}}{d_2} \right) \tag{8}$$

For 31 samples in the population, d_2 is equal to 4.113 [27]. Finally, the state of each edge will be determined according to the previous equations 6, 7, and 8. This procedure is also shown in algorithm 1.

- $UCL = 14.225 + (3) \left(\frac{10.51}{4.113} \right) \approx 22$
- $CL = 14.225$
- $LCL = 14.225 - (3) \left(\frac{10.51}{4.113} \right) \approx 7$

After detecting the statistical stability thresholds, we redefine the descriptive reference process model in the following.

Definition 9 Descriptive reference process model *V2* (\mathcal{P}) The updated definition of the descriptive reference process model or the common behaviors will contain *activities* (\mathcal{A}) and *edges* (\mathcal{E}) that respect the following conditions:

$$\begin{aligned} & [[\forall \mathcal{A} \exists s_a] \wedge [\forall s_a \subseteq \mathcal{S}_a \exists \bar{x}_{s_a}]] \wedge \\ & [[\forall \mathcal{E} \exists s_e] \wedge [\forall s_e \subseteq \mathcal{S}_e \exists x_{s_e}]] \\ & \therefore \\ & ((\mathcal{A}, \mathcal{E}) \in \mathcal{P}) \rightarrow [LCL < \bar{x}_{s_a} < UCL] \cup [UCL \leq \bar{x}_{s_a}] \wedge \\ & [LCL.edge < x_{s_e} < UCL.edge] \cup [UCL.edge \leq x_{s_e}] \end{aligned} \tag{9}$$

This definition—equation 9—expresses the conditions for the sets of *activities* (\mathcal{A}) and *edges* (\mathcal{E}) to be included within the descriptive reference process model(\mathcal{P}).

Fig. 16 Visualizing the urology process model by using the classic heuristic miner and applying a dependency threshold of 85%. The selection of the threshold value is random due to the approach of the algorithm

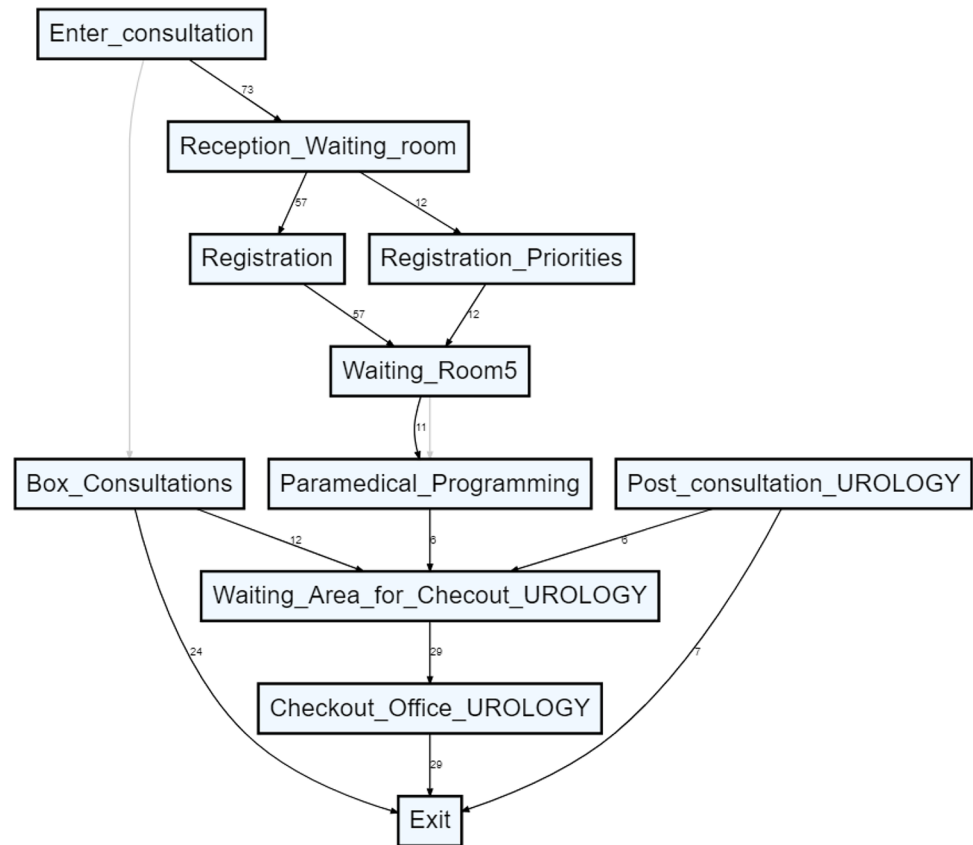


Table 3 Comparing the number of observed behaviors in the event log, with the result of the stable heuristic miner V1

	Lower than LCL	In the stable State	Higher than UCL
Number of activities	4	8	2
Total number of modeled activities		10	
Total number of observed activities (recorded in the event log)		14	

As shown, for all activities (\mathcal{A}), there exists a sample (s_a) that represents the behavior of each activity. A value as \bar{x}_{s_a} shows the average behavior of each activity.

Additionally, the behavior of all the edges (\mathcal{E}) is presented by a sample that has a value of x_{s_e} . Therefore, a set of an activity and its edges is within the descriptive reference process model (\mathcal{P}), if the average of the activity behavior (\bar{x}_{s_a}) is between the *UCL* and *LCL*. If \bar{x}_s is greater than *UCL* then the activity is considered as an activity with high instability.

Simultaneously, edges (and the linked activities) are within \mathcal{P} if their values (x_{s_e}) fall between the two thresholds *UCL.edge* and *LCL.edge*. If x_{s_e} is greater than *UCL.edge*, then this edge is considered as an edge with high instability as well.

Note that the new evaluation of *UCL.edges*, *LCL.edges*, and *CL.edges* are in parallel with the previous calculation of the first version of the algorithm. Basically, this branch of calculations is added to improve a previous shortcoming of the algorithm. The result of applying the new algorithm is presented in Fig. 8. Needless to say, the behavior of activities was discovered by using the first version of the algorithm [10, 12], which had a focus on activities' behaviors. However, to simplify our explanation we avoid re-explaining the previous work. In the following, we will compare the results of applying the *stable heuristic miner 1 & 2* and *classic heuristic miner* to the running example. This procedure is also shown in algorithm 2.

Algorithm 1 Stable Heuristic Miner V2

Fig. 17 The discovered model of the urology department by the stable heuristic miner 1. The statistical stability method is applied only to activities and behaviors

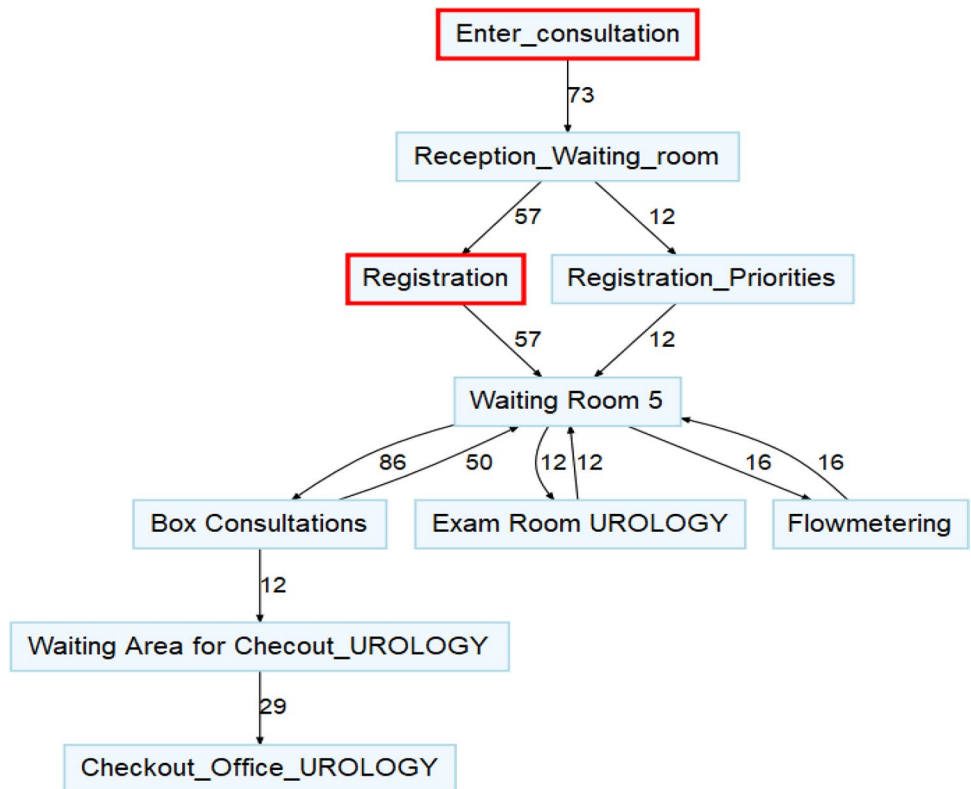
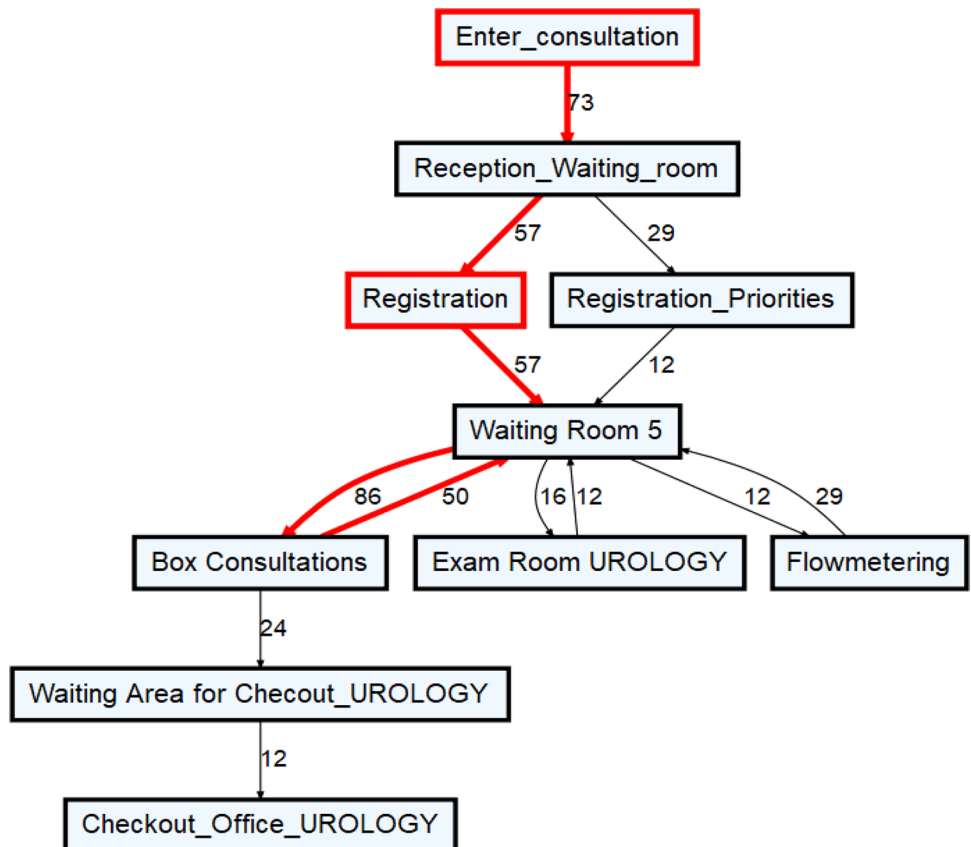


Fig. 18 The discovered model of the urology department by the stable heuristic miner V2. The Statistical stability methods are applied for both activities and edge behavior



```

1: procedure EXTRACT THE THRESHOLDS
2:   Input Footprint.Matrix
3:   Input Edges.Set
4:   Output UCL, LCL, CL
5:    $m = \text{length}(\text{Edges.Set})$  ▷ get number of edges
6:   ▷ total number of observations
7:   in Footprint.Matrix
8:     gather (columns, values, row);
9:     arrange (row, -val); ▷ sorting in a descending order
10:    filter (val !=0);
11:    create data.table1
12:  in data.table1
13:    mutate(MR = absolute( $x_i - x_{i-1}$ ));
14:    mean( $x_i$ )
15:
16:   $CL = \bar{x}$  ▷ devise the thresholds
17:
18:   $UCL = \bar{x} - 3(\frac{MR}{d_2})$ 
19:
20:   $LCL = \bar{x} + 3(\frac{MR}{d_2})$ 
21: end procedure

```

Algorithm 2 Stable Heuristic Miner V2

```

1: procedure IDENTIFY THE STATUS OF ACTIVITIES AND EDGES
2:   Input UCL, LCL, CL
3:   Input Edges.Set
4:   Input Activity.Set
5:   Input data.table1
6:   Output Graph
7:   ▷ considering the average of relative frequencies for each sample
8:   for i in data.table1[ $x_i$ ] do
9:     Nodes.Stable.Edges = unique(subset(data.table1, LCL <  $x_i$  < UCL))
10:    Nodes.Unstable.Edges = unique(subset(data.table1,  $x_i \leq LCL$ ));
11:    Node.Hot.Edges = unique(subset(data.table1, UCL ≤  $x_i$ ))
12:  end for
13:  all.nodes=match(combine(Nodes.Stable.Edges, Nodes.Unstable.Edges),
14:  Activity.Set); ▷ Select the nodes
15:  stable.edges = match(Nodes.Stable.Edges, data.table1[FirstActivity, SecondActivity]);
16:  hot.edges = match(Node.Hot.Edges, data.table1[FirstActivity, SecondActivity], Color.Attr= "red");
17:  process.edges = combine(stable.edges, hot.edges)
18:  ▷ Select the edges;
19:  process.graph= create.graph( nodes= all.nodes, edges = process.edges)
20:  render.graph(process.graph)
21: end procedure

```

3.2 Evaluating the Method on the Running Example

Figures 3, 4, 5, and 6 present the result of the classic heuristic miner for the running example. The thresholds for determining the level of dependency measures for activities in the event log are set in an arbitrary manner which is considered a disadvantage of this algorithm [7, 8].

At first, we applied some known process discovery quality criteria evaluation metrics: precision, fitness, and complexity in both size and behavior of the model [29, 30]. Previous research provided a concrete definition of these criteria [31]. *Precision* evaluates the ability of the discovered model to reproduce exactly what has been found in the original event log. This metric ranges between 0 and 1, where a value of the maximum of 1 means that the model can only generate sequences that are already seen in the event log. *Fitness*

measures to what degree the recorded behavior in the event log is represented by the process model.

Complexity evaluates the difficulty of analyzing a process model by humans. *Simplicity* is just the opposite value of complexity. One could argue that the complexity and simplicity criteria are subjective, and it is difficult to quantify a quality metric this way. In this regard, the new algorithm effectively stabilizes the metrics in the four quality dimensions.

The extracted results were evaluated by these presented metrics, which are depicted in Table 2. Looking at this evaluation and corresponding criteria, choosing a value to represent the reference behavior would be arbitrary. Therefore, deriving a conclusion to select the descriptive reference process model solely based on the traditional process discovery criteria poses a challenge.

Accordingly, the new application of *statistical stability* ensures the detection of a stable amount of information from an event log. This task will be carried out by certain *reactive thresholds*, which are determined by using the statistical stability methods. Therefore, by using the stable heuristic miner algorithm, it is not required anymore to arbitrarily select the threshold values. As a result, activities and edges with insignificant behaviors will be removed. A common path will present the stable behaviors in an event log, and the most variant behaviors will be detected as well. As stated before, these behaviors that are shown in red could raise issues in the process.

The descriptive reference process model extracted by the new algorithm is more explanatory in comparison with the classic heuristic miner and the previous stable heuristic miner. This is due to the automatic detection of stable and unstable behaviors for both edges and activities.

4 Case Studies: Discussion and Evaluation

A pragmatic approach is selected to assess the capability of the new algorithm based on real-world scenarios that are:

- LivingLabHospital_Interpreted Location event logs [15]: This location event log is included in this article for the first time by the authors. It contains the events related to the movements of patients inside the urology department of the Toulouse hospital facility in the south of France. This data was recorded by the authors to monitor and improve patients' pathways by using location data and process mining. This data is shared in this article, and a novel analysis could help compare the results of both versions. We focus on analyzing patients in the urology department. To better understand how the data was inter-

preted and prepared for process discovery actions, we refer to the works in [32, 33].

- Sepsis Data set [34]: This event log consists of sepsis cases in a hospital. There are 1000 cases with a total of 15,000 events that were recorded for a total of 16 different activities.

4.1 Comparing the Results of Different Versions of the Algorithm Based on Sepsis Data

At first, we analyzed this dataset with the classic heuristic miner. Figures 9, 10, 11 present the results. As depicted there, the classic heuristic miner results in an arbitrary selection of thresholds, and it is not efficient in extracting the reference process model.

On the other hand, Fig. 12 presents the outcome obtained by applying the initial version of the stable heuristic miner algorithm to the sepsis data set. As mentioned, this data set contains the behaviors of 16 different activities. Nonetheless, the process discovery result shows that all of these activities exhibit stable behaviors.

Within this assessment, Fig. 13 depicts the outcome of the new version of the stable heuristic miner algorithm. Based on this result, edges related to four activities represented insignificant behaviors in the statistical population. These activities were ‘Release B’, ‘Release C’, ‘Release D’, and ‘Release E’. Consequently, they have been removed. There are no activities with high instability.

Moreover, according to Fig. 13, several edges show high instability in their behaviors. Such instability could be the cause of one or several inefficiencies. For instance, consider *ER Triage* activity. Here are all the outgoing edges:

- *ER Triage* → *CRP*: 52 (frequency)
- *ER Triage* → *Leucocytes*: 29 (frequency)
- *ER Triage* → *LacticAcid*: 48 (frequency)
- *ER Triage* → *ER Sepsis Triage*: 905 (frequency)

There is one edge (*ER Triage* → *ER Sepsis Triage*) that shows a higher instability and abnormal behavior than others. This edge is shown with a ‘red’ color code in Fig. 13. Unstable behavior may signal underlying issues. With this updated algorithm, we can pinpoint potential bottlenecks and inefficiencies in our process, an analysis beyond the classic heuristic miner. To our knowledge, conventional methods lack this insight. Assessing event log stability yields a more informative process model, useful for automatic diagnosis of process deviations and concept drift detection. It should be noted that there is a difference between the in-going and the outgoing frequencies of edges. This is due to removing the insignificant edges (edges with values lower than *LCL*). Furthermore, the stable heuristic miner algorithm efficiently handles large event logs,

like sepsis data, due to improved statistical methods for large datasets.

4.2 Comparing the Results of Different Versions of the Algorithm Based on Urology Data

As indicated earlier, this case study was designed to detect inefficiencies in patients’ pathways. Indoor localization systems were used to collect this data.

Similarly in this case study, firstly, we evaluated the results by applying the classic heuristic miner. Figures 14, 15, 16 depict the results. As expected, the results failed to provide a reference process model according to solid mathematical logic.

On the contrary, Fig. 17 presents the effort to detect a reference model by the first version of the stable heuristic miner algorithm. Out of the 14 activities recorded in the event log, 10 of them are shown by the result of the first version of the stable heuristic miner algorithm. Table 3 summarizes this observation.

Eight of the ten activities expressed stable behaviors, and two of them showed high instability in comparison with the total number of recorded behaviors. However, the *statistical stability of edges are not addressed* here.

Figure 18 illustrates the result of the second version of the stable heuristic miner algorithms by using the urology patients’ data.

By means of this algorithm, not only unstable activities but unstable edges are discovered as well. Now, by considering the “*Registration*” activity, it is possible to recognize the source of instability in this department.

The high level of fluctuations caused by this activity is seen by the connection/edge between “*Registration*” and “*Reception_ Waiting_ room*”. The value of these edges is 57, which is higher than the normal value for all the other registered behaviors. As a result, it puts the process in an *unstable state*.

The same behavior causes instability for edges between “*Waiting_Room5*” and “*Box Consultation*” activities.

Detection of these deviating behaviors could help the domain experts in highlighting the causes of inefficiencies in patients’ pathways.

It is important to stress that acquiring such a diagnosis is only feasible if experts are assured of a mathematical logic behind the process discovery. A logic that could lead to the extraction of the descriptive reference process model of patients’ pathways. Based on our extensive experiments involving the monitoring of patients’ activities, it was evident that the outcome observed was not attainable using the classic heuristic miner algorithm.

Moreover, the novel approach of the stable heuristic miner algorithm helped experts to *detect deviating behaviors*

automatically and to capture an image of what patients normally do, even if the domain experts did not particularly have a piece of complete knowledge about the process.

In the past, domain experts were required to possess an in-depth understanding of the process, as the discovered model had to be manually filtered. However, given the intricate nature of complex healthcare processes, this task posed significant challenges.

5 Conclusion

Heuristic-based process discovery methods showed promising results in the literature on process mining. For instance, based on the literature review in [1, 3], the classic heuristic miner algorithm has been identified as the most applied process discovery algorithm in the healthcare sector. The heuristic-based methods discover an initial set of activities and edges according to dependency measures and then modify the set regarding some arbitrarily defined thresholds. This has been identified in this article and the literature as a scientific challenge leading to the extraction of a non-optimal set of activities and edges.

Based on the hypothesis of this article, evaluating the statistical stability in an event log could lead to the discovery of a descriptive reference process model. Eventually, this evaluation could address the mentioned scientific gap. This hypothesis is mainly driven by the definition of statistical stability phenomenon that addresses the question of “*how one can represent the reference behavior of a complex system with emerging properties*”. Such a workflow could be used as one of the building blocks of data-driven simulation models.

The first version of the stable heuristic miner algorithm introduced the evaluation of statistical stability in an event log. However, the assessment was carried out only by considering activities’ behaviors. This has been mentioned as a limitation of the previous version of the algorithm.

In this article, we presented a new version of the stable heuristic miner. We redefined the descriptive reference process model and presented an assessment of this algorithm by using two real-life event logs. Thanks to this algorithm, we were able to extract a more comprehensive and informative descriptive reference process model. The discovered results showed the detection of unstable and insignificant behaviors for both activities and edges among them. Accordingly, users could obtain a stable amount of information from an event log *without the need to manually and arbitrarily modify a filtering value*. The obtained model could also detect a stable state among 4 traditional process discovery quality dimensions criteria (i.e., fitness, precision, generalization, and simplicity/complexity).

5.1 Limitations and Challenges

It is a demanding task to evaluate the results of a process discovery algorithm. Some researchers used conformance-checking-based methods to identify and evaluate certain criteria such as fitness, simplicity, generality, etc. [1]. These criteria have been applied in this article for the running example. They evaluate the correlation between the recorded information in the event log and the discovered model. However, these methods are not the most effective metrics to measure statistical stability results. The nature of this work is based on the assessment of statistical stability criteria. Accordingly, methods and definitions are presented to examine and eventually discover the behavior registered in event logs. Assessing statistical stability performed by conformance checking-based criteria is conceived as not the best practice. Similar issues and concerns have been seen before in other research works as well [7, 31, 35]. We aim to address this challenge in our future research, with the objective of developing an overall applicable evaluation approach.

5.2 Future Perspective

We are planning to focus on devising a set of quantitative evaluation criteria for this algorithm rather than a qualitative approach. Moreover, this algorithm has the potential for automatic diagnosis and reasoning of business processes. Accordingly, we are aiming to improve the task of process discovery by developing a knowledge-driven approach. Also, it is beneficial to integrate a procedural semantic in the representation of the process discovery results rather than a declarative semantic. The other avenue of research is the application of this algorithm for the discovery of a data-driven simulation model.

Acknowledgements The authors would like to thank all reviewers of the manuscript for their input.

Author Contributions Evaluation of the statistical stability phenomenon for both activities and edges in an event log. Introducing the stable heuristic miner 2 process discovery algorithm. Automatic detection of deviating behaviors in an event log. A potent method to be used for concept drift detection. Conceptualization: Sina NAMAKI ARAGHI. Methodology: Sina NAMAKI ARAGHI, Frederick BENABEN, Franck FONTANILI. Software: Sina NAMAKI ARAGHI. Validation: Sina NAMAKI ARAGHI, Frederick BENABEN, Franck FONTANILI, Elyes LAMINE. Resources: Frederick BENABEN, Franck FONTANILI. Writing original draft: Sina NAMAKI ARAGHI. Writing—review: Frederick BENABEN, Elyes LAMINE. Project Administration: Frederick BENABEN, Franck FONTANILI.

Funding Not applicable.

Availability of Data and Materials The data of the mentioned case study will be shared in this publication [15].

Declarations

Conflict of interest The authors have no competing interests to declare that are relevant to the content of this article.

Consent for Publication The authors hereby consent to the publication of the work.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

References

- Augusto A, Conforti R, Dumas M, Rosa ML, Maggi FM, Marrella A, Mecella M, Soo A. Automated discovery of process models from event logs: review and Benchmark. *IEEE Trans Knowl Data Eng.* 2018. <https://doi.org/10.1109/TKDE.2018.2841877>.
- Garcia CS, Meincheim A, Faria ER Jr, Dallagassa MR, Sato DMV, Carvalho DR, Santos EAP, Scalabrin EE. Process mining techniques and applications—a systematic mapping study. *Expert Syst Appl.* 2019;133:260–95. <https://doi.org/10.1016/j.eswa.2019.05.003>.
- Rojas E, Munoz-Gama J, Sepúlveda M, Capurro D. Process mining in healthcare: a literature review. *J Biomed Inf.* 2016;61:224–36. <https://doi.org/10.1016/j.jbi.2016.04.007>.
- Weijters AJMM, van der Aalst WMP, Alves De Medeiros AK. Process Mining with the HeuristicsMiner Algorithm. BETA publicatie : working papers. Technische Universiteit Eindhoven; 2006.
- Weijters AJMM, Ribeiro JTS. Flexible Heuristics Miner (FHM). In: 2011 IEEE symposium on computational intelligence and data mining (CIDM); 2011. pp. 310–317. <https://doi.org/10.1109/CIDM.2011.5949453>.
- Bakhshi A, Hassannayebi E, Sadeghi AH. Optimizing sepsis care through heuristics methods in process mining: a trajectory analysis. *Healthc Anal.* 2023;3: 100187. <https://doi.org/10.1016/j.health.2023.100187>.
- De Cnudde S, Claes J, Poels G. Improving the quality of the heuristics miner in ProM 6.2. *Expert Syst Appl.* 2014;41(17):7678–90. <https://doi.org/10.1016/j.eswa.2014.05.055>.
- Tavakoli-Zaniani M, Gholamian MR, Golpayegani SAH. Improving heuristic-based process discovery methods by detecting optimal dependency graphs; 2022. <https://doi.org/10.48550/arXiv.2203.10145>, [arXiv:2203.10145](https://arxiv.org/abs/2203.10145) [cs]. Accessed 2022-11-30.
- Kurniati A, Kusuma GP, Wisudiawan G. Implementing Heuristic Miner for Different Types of Event Logs. 2016. <https://www.semanticscholar.org/paper/Implementing-Heuristic-Miner-for-Different-Types-of-Kurniati-Kusuma/417a14e5aefdb42711d98cfeabcf5ccec6ada299> Accessed 2023-01-12.
- Namaki Araghi S, Fontanili F, Lamine E, Okongwu U, Benaben F. Stable heuristic miner: applying statistical stability to discover the common patient pathways from location event logs. *Intell Syst Appl.* 2022;14: 200071. <https://doi.org/10.1016/j.iswa.2022.200071>.
- Namaki Araghi S, Fontanili F, Sarkar A, Lamine E, Karray M-H, Benaben F. Diag approach: introducing the cognitive process mining by an ontology-driven approach to diagnose and explain concept drifts. *Modelling.* 2024;5(1):85–98. <https://doi.org/10.3390/modelling5010006>.
- Namaki Araghi S. A methodology for business process discovery and diagnosis based on indoor location data: Application to patient pathways improvement. These de doctorat, Ecole nationale des Mines d'Albi-Carmaux (November 2019). <https://www.theses.fr/2019EMAC0014> Accessed 2023-09-20.
- Gorban II. Phenomenon of statistical stability. *Tech Phys.* 2014;59(3):333–40. <https://doi.org/10.1134/S1063784214030128>.
- Gorban II. The statistical stability phenomenon. *Math Eng.* 2017. <https://doi.org/10.1007/978-3-319-43585-5>.
- Namaki Araghi S. LivingLabHospital_interpreted Location event logs. 2023;1. <https://doi.org/10.17632/v5kc7chhpv.1>. Publisher: Mendeley Data. Accessed 2023-06-05.
- Aalst WMP: Data science in action. Berlin, Heidelberg: Springer; 2016. pp. 3–23. https://doi.org/10.1007/978-3-662-49851-4_1.
- Burattin A, Sperduti A, Aalst WMP. Heuristics Miners for Streaming Event Data. In: 2014 IEEE Congress on Evolutionary Computation (CEC); 2014. pp. 2420–2427. <https://doi.org/10.1109/CEC.2014.6900341>. [arXiv:1212.6383](https://arxiv.org/abs/1212.6383) [cs]. Accessed 2023-01-13.
- Burattin A. Process mining techniques in business environments. Lecture notes in business information processing, vol. 207. Cham: Springer; 2015. <https://doi.org/10.1007/978-3-319-17482-2>. <http://link.springer.com/10.1007/978-3-319-17482-2> Accessed 2023-01-13.
- Broucke SKLM, Weerd JD. Fodina: a robust and flexible heuristic process discovery technique. *Decis Support Syst.* 2017;100:109–18. <https://doi.org/10.1016/j.dss.2017.04.005>.
- Prodel M. Process discovery, analysis and simulation of clinical pathways using health-care data. Theses, Université de Lyon; April 2017. <https://theses.hal.science/tel-01665163>.
- Werf JMEM, Dongen BF, Hurkens CAJ, Srebenik A. Process discovery using integer linear programming. In: Hee KM, Valk R. editors. Applications and theory of petri nets. Lecture Notes in Computer Science, Berlin, Heidelberg: Springer; 2008. pp. 368–387. https://doi.org/10.1007/978-3-540-68746-7_24.
- Yahya BN, Song M, Bae H, Sul S-O, Wu J-Z. Domain-driven actionable process model discovery. *Comput Ind Eng.* 2016;99:382–400. <https://doi.org/10.1016/j.cie.2016.05.010>.
- Prodel M, Augusto V, Jouaneton B, Lamarsalle L, Xie X. Optimal Process Mining for Large and Complex Event Logs. *IEEE Trans Autom Sci Eng.* 2018;15(3):1309–25. <https://doi.org/10.1109/TASE.2017.2784436>.
- Zelst SJ, Dongen BF, Aalst WMP, Verbeek HMW. Discovering workflow nets using integer linear programming. *Computing.* 2018;100(5):529–56. <https://doi.org/10.1007/s00607-017-0582-5>.
- Leemans SJJ, Fahland D, van der Aalst WMP. Process and deviation exploration with inductive visual miner. In: Limonad L, Weber B. editors. BPM Demo Sessions 2014 (co-located with BPM 2014, Eindhoven, The Netherlands, September 20, 2014). CEUR Workshop Proceedings, pp. 46–50. CEUR-WS.org; 2014. BPM Demo Sessions 2014 (BPMD 2014), September 10, 2014, Eindhoven, The Netherlands, BPMD 2014 ; Conference date: 10-09-2014 Through 10-09-2014.
- Augusto A, Conforti R, Dumas M, Rosa ML, Maggi FM, Marrella A, Mecella M, Soo A. Automated discovery of process models from event logs: review and benchmark. *IEEE Trans Knowl Data*

- Eng. 2019;31(4):686–705. <https://doi.org/10.1109/TKDE.2018.2841877>.
27. Montgomery DC. Introduction to Statistical Quality Control, 8th edn. Industrial Engineering/Manufacturing. General and Introductory Industrial Engineering. Subjects. Wiley; 2007. <https://www.wiley.com/en-us/Introduction+to+Statistical+Quality+Control>. Accessed 2019-08-29.
 28. Introduction to Statistical Quality Control, 8th edn. Wiley. <https://www.wiley.com/en-us/Introduction+to+Statistical+Quality+Control> Accessed 2023-01-13.
 29. Buijs JCAM, Dongen BF, Aalst WMP. Quality dimensions in process discovery: the importance of fitness precision generalization and simplicity. *Int J Cooper Inf Syst*. 2014;23(1):144. <https://doi.org/10.1142/S0218843014400012>.
 30. Janssenswillen G, Donders N, Jouck T, Depaire B. A comparative study of existing quality measures for process discovery. *Inf Syst*. 2017;71:1–15. <https://doi.org/10.1016/j.is.2017.06.002>.
 31. The connection between process complexity of event sequences and models discovered by process mining. Elsevier Enhanced Reader. <https://doi.org/10.1016/j.ins.2022.03.072>.
 32. Araghi SN, Fontaili F, Lamine E, Salatge N, Lesbegueries J, Pouyade SR, Tancerel L, Benaben F. A conceptual framework to support discovering of patients' pathways as operational process charts. In: 2018 IEEE/ACS 15th international conference on computer systems and applications (AICCSA), IEEE; 2018. pp. 1–6.
 33. Namaki Araghi S, Fontanili F, Lamine E, Salatge N, Benaben F. Interpretation of Patients' Location Data to Support the Application of Process Mining Notations. In: HEALTHINF 2020 - 13th International Conference on Health Informatics. Proceedings of the 13th International Joint Conference on Biomedical Engineering Systems and Technologies - HEALTHINF, vol. 5; 2020. pp. 472–481. SCITEPRESS - Science and Technology Publications, La Valette, Malta. <https://doi.org/10.5220/0008971104720481>.
 34. eventdataR/.Rhistory at master · gertjanssenswillen/eventdataR. <https://github.com/gertjanssenswillen/eventdataR> Accessed 2022-12-07.
 35. Estrada-Torres B, Camargo M, Dumas M, García-Bañuelos L, Mahdy I, Yerokhin M. Discovering business process simulation models in the presence of multitasking and availability constraints. *Data Knowl Eng*. 2021;134: 101897. <https://doi.org/10.1016/j.datak.2021.101897>.

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.