From Action to Response to Effect: Mining Statistical Relations in Work Processes

Jelmer J. Koorn[a,*], Xixi Lu[a], Henrik Leopold[b,c], Hajo A. Reijers[a]

[a] *Utrecht University, Princetonplein 5, Utrecht, The Netherlands*
[b] *Kühne Logistics University, Großer Grasbrook 17, Hamburg, Germany*
[c] *Hasso Plattner Institute, Prof.-Dr.-Helmert-Straße 2-3, University of Potsdam, Germany*

Abstract

Process mining techniques are valuable to gain insights into and help improve (work) processes. Many of these techniques focus on the sequential order in which activities are performed. Few of these techniques consider the statistical relations within processes. In particular, existing techniques do not allow insights into how responses to an event (action) result in desired or undesired outcomes (effects). We propose and formalize the ARE miner, a novel technique that allows us to analyze and understand these action-response-effect patterns. We take a statistical approach to uncover potential dependency relations in these patterns. The goal of this research is to generate processes that are: 1) appropriately represented, and 2) effectively filtered to show meaningful relations. We evaluate the ARE miner in two ways. First, we use an artificial data set to demonstrate the effectiveness of the ARE miner compared to two traditional process-oriented approaches. Second, we apply the ARE miner to a real-world data set from a Dutch healthcare institution. We show that the ARE miner generates comprehensible representations that lead to informative insights into statistical relations between actions, responses, and effects.

Keywords: `process discovery, statistical process mining, effect measurement`

1. Introduction

Process mining is a family of techniques that helps organizations to understand, analyze, and improve their work process [1, 2]. The basis for process mining techniques and their analyses are so-called event logs. These event logs are extracted from various information systems that are used within the organizations and, therefore, provide valuable insights into how work processes are actually executed [3].

Although the value of process mining has been demonstrated in various contexts, its application is still associated with a number of challenges [2, 4]. Two key aspects concern the way the process mining results are presented to the user. The representation of the results must be 1) easy to understand and 2) allow the user to obtain the required insights into the process execution. In the past, various process discovery techniques have been proposed for this purpose including the heuristic miner [5], the fuzzy miner [6], and the inductive miner [7]. These techniques, however, all approach process discovery from a control-flow perspective, i.e., they discover ordering constraints among events.

Depending on the context, the control-flow perspective is not sufficient for understanding all relevant aspects of the process execution. Consider, for example, a care process in a residential care facility. In such a facility, clients with mental and/or physical disabilities reside and the care staff supports these clients in their daily lives. The main goal of most processes in this facility is to ensure the well-being of clients. To that end, the facility, among others, aims to minimize the aggressive behavior that is prevalent at the

*Corresponding author
Email address: `j.j.koorn@uu.nl` (Jelmer J. Koorn)

facility since it negatively impacts the well-being of the clients and the staff. Aggressive behavior can take many forms. For example, clients can become verbally aggressive or physically attack other clients, staff, or also themselves. When a client becomes aggressive, a staff member responds to the aggressive incident using one or multiple measures. These measures range from mild measures, such as verbal warnings, to severe measures, such as seclusion. The care facility is particularly interested in uncovering which of these measures lead to desired (i.e., de-escalation of aggressive behavior) or undesired (i.e., escalation of aggressive behavior) outcomes.

To understand and analyze such a process, we need to identify *action-response-effect* patterns. In our care process example, the aggressive incident of the client represents an *action*, the measure taken by the staff is a *response*, and the future behavior of the client is the *effect*. Existing process mining techniques cannot identify such action-response-effect patterns since their discovery requires an analysis beyond the control-flow perspective. If we were to apply existing discovery techniques to an event log from such a care process, this would result in an unsatisfactory process representation for two reasons. First, the representation would be hard to read because it would contain too many connections. Second, the representation would not allow the organization to obtain the insights they require because the resulting representation would not show the effect of the behavior.

In light of these limitations of existing techniques, we propose a novel discovery technique in this work: the ARE miner. For this technique, we take advantage of well-established statistical tests to analyze event logs. The goal of this analysis is to discover and visualize understandable graphical representations of work processes. We achieve this by highlighting statistically significant and hiding the statistically insignificant relations that we discover through the statistical tests. In order to investigate the effectiveness of the technique, we evaluate it on an artificial data set and compare the results to a technique from the control flow-perspective: the directly-follows graph. Furthermore, we demonstrate the applicability of the technique by conducting a case study in a Dutch residential care facility. We analyze a total of 21,384 aggression incidents related to 1,115 clients. Combining the insights from these two evaluations, we show that the ARE miner provides graphical representations that are 1) easy to understand, and 2) highlight informative insights.

This work is an extension of our earlier work that was published in the proceedings of the 18th International Conference on Business Process Management [8]. We extended the original paper significantly in various ways. There are six main differences: 1) improved graphical representation of arcs, 2) automated determination of conceptual parameter epsilon, 3) introduction of a comprehensive quantitative evaluation, 4) extension of the qualitative evaluation, 5) revision of the related work, and 6) a thorough discussion of the limitations. We improved the *graphical representation* by including the strength of the identified statistical relations in the representation. As a further refinement of our technique, we introduced an approach to *automatically determine the parameter epsilon*, which was formerly done manually by domain experts. Now we included a data driven approach to increase the generalizability and applicability of our technique. Besides these conceptual differences, we also extended the evaluation. We conducted a comprehensive *quantitative evaluation* based on an artificial data set to demonstrate the performance of the ARE miner in a broad spectrum of contexts. Furthermore, we extended the *qualitative evaluation* in two ways. First, we included and discussed all graphical representations from the case study to provide a more comprehensive view of the results. Second, we increased the depth by discussing additional types of relations. We revised the *related work* part by including a discussion on causal process mining techniques and by analyzing the differences and overlaps between existing techniques and the ARE miner. Finally, we expanded the *limitations* by critically reflecting on both limitations of the ARE miner itself as well as its evaluation.

The rest of the paper is organized as follows. Section 2 describes and exemplifies the problem of discovering *action-response-effect* patterns. Section 3 introduces the formal preliminaries for our work. Section 4 describes the ARE miner for discovering *action-response-effect* patterns. Section 5 presents the evaluation of the ARE miner based on an artificial data set and a real-world event log. Section 6 elaborates on the insights, implications, and limitations of our work. Section 7 discusses related work before Section 8 concludes the paper.

EID	CID	Timestamp	Action	Response
1	1	12-05 09:53	VA	Warning
2	1	13-05 13:35	PO	Distract Client, Seclusion
3	1	26-05 09:32	VA	Warning
4	1	26-05 11:02	PP	Distract Client
5	2	21-06 14:51	VA	Distract Client
6	1	23-06 21:23	VA	Distract Client
7	2	24-06 17:02	VA	-
8	3	29-08 11:22	VA	Warning
9	3	31-08 08:13	PO	Warning, Seclusion
10	3	31-08 10:48	PP	Distract Client

Legend: EID = Event identifier, CID = Client identifier, VA = Verbal Aggression, PP = Physical Aggression People), PO = Physical Aggression (Objects

Table 1: Excerpt from an action-response log of a care process

2. Problem Statement

Many processes contain action-response-effect patterns. As examples consider healthcare processes where doctors respond to medical conditions with a number of alternative treatments, service processes where service desk employees respond to issues with technical solutions, and marketing processes where customers may respond to certain stimuli such as ad e-mails with increased demand. Let us reconsider the example of the healthcare process in a residential care facility in order to illustrate the challenge of discovering an understandable and informative process representation from an event log containing action-response relations. Of particular interest are the incidents of aggressive behavior from the clients and how these are handled by staff. Table 1 shows an excerpt from a respective event log. Each entry consists of:

1. an event identifier EID (which, in this case, is equal to the incident number),
2. a case identifier CID (which, in this case, is equal to the client identifier),
3. a timestamp,
4. an aggressive incident (action),
5. one or more responses to this event.

Figure 1 a) shows the directly–follows graph that can be derived from the events of this log. It does not suggest any clear structure of the process. Although this graph is only based on twelve events belonging to three different event classes, it seems that almost any behavior is possible. In addition, this representation does not provide any insights into certain hidden patterns [9]. However, if we take a closer look, we can see that there are effects for a certain response. For instance, we can see that, over time, the aggressive incidents related to client 1 escalate from verbal aggression to physical aggression against objects and people. The verbal aggression event in June (EID = 6) is probably unrelated to the previous pattern since it occurs several weeks after. To gain an even deeper understanding, we need to take both the response and its effect into account. When we consider this, we see that both client 1 and 2 escalate from verbal aggression to physical aggression after the verbal aggression was only countered with a warning.

These examples illustrate that explicitly including the responses and effects in the discovery phase is important for answering the question of how to possibly respond to an action when a certain effect (e.g., de-escalating aggressive behavior) is desired. Therefore, our objective is to discover a model that: (1) shows the action-response-effect process, and (2) reveals the statistical patterns of which responses lead to a desired or undesired outcome (effect). There are two main challenges associated with accomplishing this:

1. *Graphical representation*: From a control-flow perspective, action-response relations are a loop consisting of a choice between all actions and a subsequent and-split that allows to execute or skip each response. Figure 2 b) illustrates this by showing the Petri net representing the behavior from the log in Table 1. Obviously, this representation does not allow to understand which responses lead to a desired or undesired effect.

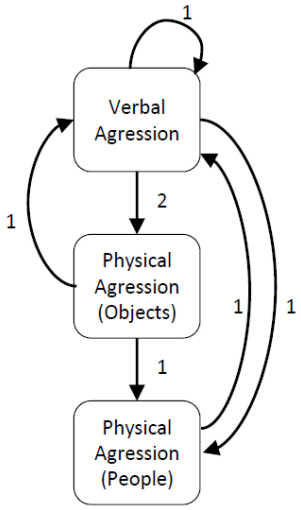

Figure 1: Directly-follows graph

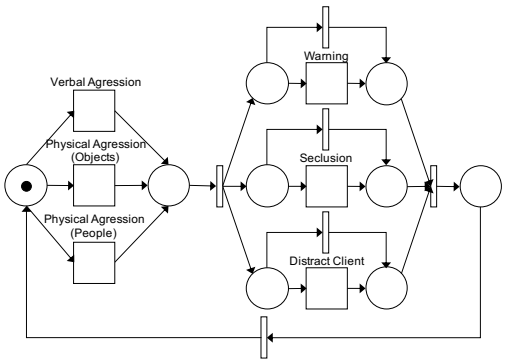

Figure 2: Petri Net

2. *Effective filtering mechanism*: The possible number of responses calls for a filtering mechanism that allows to infer meaningful insights from the model. In the example above, we only have three event classes and three event response classes (plus the "no response" class). This results in eight possible responses. In case of 5 response event classes, we already face 32 (=2^5 possible responses classes). Including all these response arcs in a model will likely lead to an unreadable model that does not allow to provide the desired insights.

In the next sections, we propose a novel technique, the ARE miner, that creates graphical representations of statistical patterns in event logs that contain actions, responses, and effects.

3. Preliminaries

As discussed, action-responses-effect patterns are observed in many processes and can provide diagnostic information regarding the follow-up effects of responses. To discover these patterns, we build on the well-established concept of event logs and introduce the concept *action-response-effect logs*. In this section, we first formalize the action-response-effect logs. We then discuss how the effects of events are defined.

3.1. Action-Response-Effect Logs

Starting from the event logs, we follow the definition that an *event log* is a set of sequences of events being recorded. Each *sequence* registers the execution of a case, also called a *trace*, and each *event* of the sequence represents an activity executed for the same case. Moreover, each event is associated with a set of *attributes*, which provides information such as who executed this event, when is the event executed, and etc. To define action-responses-effect logs, we follow the concept of an event log and associate each event with the action (e.g., activities occurred), its response, and the effects by explicitly defining these attributes $_{l}$; $_{r}$, and $_{next}$, respectively. We formalize action-response-effect logs as follows.

Definition 1 (action-response-effect Log). Let E be the universe of event identifiers. Let C be the universe of case identifiers. Let $d_1; \ldots; d_n$ be the set of attribute names (e.g., timestamp, resource, location). Let A be the set of actions and R a finite set of responses. An action-response-effect log L is defined as $L = (E;\ _{c};\ _{l};\ _{r};\ _{next};\ _{d_1};\ldots;\ _{d_n}; <)$, where

ID	Timestamp	Action	Response	E ect
1	12-05 09:53	VA	Warning	PO
1	13-05 13:35	PO	Distract Client, Seclusion	
1	26-05 09:32	VA	Warning	PP
1	26-05 11:02	PP	Distract Client	
2	21-06 14:51	VA	Distract Client	VA
1	23-06 21:23	VA	Distract Client	
2	24-06 17:02	VA	-	
3	29-07 11:22	VA	Warning	PO
3	31-07 08:13	PO	Warning, Seclusion	PP
3	31-07 10:48	PP	Distract Client	

Legend: VA = Verbal Aggression, PP = Physical Aggression (People), PO = Physical Aggression (Objects)

Table 2: Excerpt of the event log action-response-effect

- E E is the set of events,
- $_c : E \; ! \; C$ is a surjective function linking events to cases,
- $_l : E \; ! \; A$ is a surjective function linking events to actions,
- $_r : E \; ! \; 2^R$ is a surjective function linking events to a set of responses,
- $_{next} : E \; ! \; C$ is a surjective function linking events to the effects,
- $_{d_i} : E \; ! \; U$ is a surjective function linking the attribute d_i of each event to its value,
- $<$ E E is a strict total ordering over the events.

Given an action-response-effect log L according to Definition 1, we shall use the shorthand notation $= he_1; : : : ; e_n i$ in the remainder of this paper to refer to a trace that consists of n events with an identical case identifier. Furthermore, for any pair of events e_i and e_j with $i < j$, it holds that $e_i < e_j$ according to the strict total ordering of the events in log L.

The set of response events $fr^e_1; : : : ; r^e_n g$ of an event e is given by the function $_r$; we write $_r(e) = fr^e_1; : : : ; r^e_n g$. For each trace $= he_1; : : : ; e_n i$, the sequence of responses is $h \; _r(e_1); : : : ; \; _r(e_n)i$. For example, in the action-response-effect log listed in Table 2, for event e_1: $_c(e_1) = 1$ is the case of event e_1, $_l(e_1) =$ "Verbal Aggression" (VA) is the action of e_1, and $_r(e_1) = f$"Warning"g is the set of responses of e_1.

3.2. Effects of Responses

As we discussed, we aim to investigate whether a certain response to an action has an effect on the follow-up event. As such, we measure the effectiveness of a response to an action by studying the effect. For this aim, we first define the effects of events by using function $_{next}$ and introduce parameter $$ for elapsed time. For each trace $= he_1; : : : ; e_n i$, we define the effect for each e_i, where $1 \quad i < n$ as follows: if the elapsed time to the next event e_{i+1} is less than $$, the effect $_{next}(e_i)$ of e_i is the action of e_{i+1}, else we say that the effect is a silent action $$. Formally, if $_{time}(e_{i+1}) \quad _{time}(e_i) \quad$, then $_{next}(e_i) := \; _l(e_{i+1})$, else $_{next}(e_i) := \; .$

To test the hypothesis whether an effect is independent of the response to an action, the number of observed events is compared to the number of expected events of different responses and effects. To calculate the number of observed events, we create a matrix (table) where each cell is filled with the number of observed events of a response and an effect. Let $a \; 2 \; A$ be an action, $R = fr_1; \quad ; r_m g$ be a set of responses, and $C = fc_1; \quad ; c_n g$ a set of effects. We define a $jRj \quad jCj$ matrix, where each row represents a response r_i, each column represents an effect c_j, and each cell counts the number of observed events that have response r_i and effect c_j. We have

Observed	**PO**	**PP**	**VA**		*Total*
Warning	250	400	200	50	*900*
Held with force	20	50	50	10	*130*
Seclusion	30	50	20	10	*110*
Terminate contact	100	100	90	10	*300*
Distract client	100	150	40	20	*310*
Total	*500*	*750*	*400*	*100*	**1750**
Expected	**PO**	**PP**	**VA**		*Total*
Warning	257.1	385.7	205.7	51.4	*900*
Held with force	37.1	55.7	29.7	7.4	*130*
Seclusion	31.4	47.1	25.1	6.3	*110*
Terminate contact	85.7	128.6	68.6	17.1	*300*
Distract client	88.6	132.9	70.9	17.7	*310*
Total	*500*	*750*	*400*	*100*	**1750**

Legend: VA = Verbal Aggression, PP = Physical Aggression (People), PO = Physical Aggression (Objects)

Table 3: Excerpt of the tables used to perform high-level statistical tests; horizontal categories: effect, vertical categories: response

Observed	**PO**	**PP**	**VA**		*Total*
Terminate contact = 0	300	500	210	90	*1100*
Terminate contact = 1	100	100	90	10	*300*
Total	*400*	*600*	*300*	*100*	*1400*
Expected	**PO**	**PP**	**VA**		*Total*
Terminate contact = 0	314.3	471.4	235.7	78.6	*1100*
Terminate contact = 1	85.7	128.6	64.3	21.4	*300*
Total	*400*	*600*	*300*	*100*	*1400*

Legend: VA = Verbal Aggression, PP = Physical Aggression (People), PO = Physical Aggression (Objects)

Table 4: Excerpt of the tables for an individual response used to perform statistical tests; horizontal categories: effect

$$freq_{a;R;C} = \begin{matrix} 0 & & & & 1 \\ f_{1;1} & f_{1;2} & & f_{1;n} & \\ f_{2;1} & f_{2;2} & & f_{2;n} & \\ \vdots & \vdots & \ddots & \vdots & \\ f_{m;1} & f_{m;2} & & f_{m;n} & \end{matrix}$$

where
$$f_{i;j} = freq_L(a; r_i; c_j) = \quad jfe\ 2\ L\ j\quad _l(e) = a \wedge r_i\ 2\quad _r(e) \wedge \quad _{next}(e) = c_j gj \tag{1}$$

For instance, given a log L as listed in Table 2, $freq_L$("VA"; "Warning"; "PO") $= jfe_1; e_8gj = 2$. Considering Table 3 and omitting the column totals and row totals, it exemplifies a matrix $freq_{a;R;C}$. If the effects are independent of responses, then we should observe that the distribution of effects of a response is similar to the *total distribution*.

Each row r_i presents the distribution of effects $c_1; :::; c_k$ to the response r_i. To test whether each individual response r_i has an influence on the effects, we define $freq_{a;r;C}$ as a $2 \quad jCj$ matrix:

$$freq_{a;r;C} = \begin{matrix} f_{1;1} & f_{1;2} & & f_{1;n} \\ f_{2;1} & f_{2;2} & & f_{2;n} \end{matrix} \tag{2}$$

where $f_{1;j} = freq_L(a; r; c_j)$ and $f_{2;j} = \quad jfe\ 2\ L\ j\quad _l(e) = a \wedge r\ 2\quad _r(e) \wedge \quad _{next}(e) = c_j gj$.

An example of $freq_{a;r;C}$ where r is "Terminate contact" is listed in Table 4. In the following section, we describe that in our ARE miner first a chi-squared test is carried out. This allows us to calculate

Algorithm 1 Compute graph

Input: Event log L
Output: Graph $G = (V; \prec)$
{*STAGE 1: High-Level Statistics*}
for $a \in A$ **do**
  Initiate matrix $O[a] \leftarrow freq_{a;R;C}$ {*see Equation 2, calculate the observed values*}
  Compute matrix $E[a]$ {*calculate the expected values by following the chi-square test, see [10]*}
  Compute $^2_a = \frac{(O[a]-E[a])^2}{E[a]}$ {*To test the dependence between responses R and effects $A \cup \{\ \}$*}
  if ^2_a is significant **then**
    {*$O[a]$ differs from $E[a]$, thus responses R have a statistically significant influence on the effects C*}
    {*STAGE 2: Detailed Statistics*}
    for response $r \in R$ **do**
      Compute matrix $O[a]_r$, $E[a]_r$, and $^2_{a;r}$
      if $^2_{a;r}$ is significant **then**
        {*STAGE 3: Influential Points*}
        Compute adjusted standardized residuals ASR_c {*see Section 4.4*}
        for effects $c \in A \cup \{\ \}$ **do**
          if ASR_c is significant **then**
            {*draw the arc from r to c*}
            $V \leftarrow V \cup \{a_s\} \cup \{r\}$, $\prec \leftarrow \prec \cup \{(a_s; r)\} \cup \{(r; c)\}$
          end if
        end for {*effect*}
      else
        {$^2_{a;r}$ *is insignificant, i.e., r has no significant influence on C. We do not draw node r or any arc from r to C*}
      end if
    end for {*response*}
  else
    {*Observed $O[a]$ follows the expected values $E[a]$, thus response R has no statistically significant influence on the effects C; thus, no arcs are drawn*}
  end if
end for {*action*}
return G

the expected values and test the statistical dependency between responses and effects. The chi-square test compares the observed frequencies to the expected frequencies. If they differ significantly, then the null hypothesis is rejected, which means we cannot rule out that there is a statistical dependency relation between the response and the effect.

The complete event logs containing action-response-effect are used in the ARE miner that is proposed in this paper. The next section elaborates on this.

4. ARE Miner

Based on the formalization introduced the previous section, we use this section to propose the ARE miner as a novel discovery technique. The goal of the ARE miner is to generate understandable process representations that provide the user with the required insights into the execution of the process. First, we describe the required pre-processing steps. Then, we elaborate on the conceptual approach of the ARE miner, which consists of three main stages: 1) computing high-level statistics, 2) computing detailed statistics, and 3) identifying influential points.

4.1. Pre-processing the Event Log

We first pre-process the log to obtain the effects of responses. Since we are studying the effects of a response to an action, the duration between a response and its effects influences the likelihood of a statistical relation between the two. Let us return to our example: if there is an aggressive incident, there is a given response to this incident. However, if the next incident takes place after a long time (e.g., a year) it is unlikely that this new incident is still dependent on the response to the initial action. Thus, we use the parameter epsilon ( ), see Section 3.2. represents the maximum duration between two events in which the first event is still considered to have an effects on the second event. We can define in two ways: (1) based on the data or (2) based on the knowledge of a domain expert. If we base the on the data, we define it as equaling the average duration of the events. To ensure that outliers do not influence the average, depending

on the distribution of the data, a number of actions can be taken. For our specific example, the data is exponentially distributed. To account for this, we select 80% of the data when sorted on duration and take the mean of this subset of data. This results in an of 8.9 days. We round this up to full days (= 9) due to the granularity of our data. Other distributions in the data are likely to occur as well. One common example is that the data is normally distributed. In this case, we propose to define equaling the mean plus two standard deviations to the right (longest duration). This ensures that a subset of the data closest to the mean is captured.

We can also base the on the input of a domain expert. Consider the healthcare organization in our qualitative evaluation, see Section 5.2. We consulted with a behavioral expert in the organization. The domain expert from the organization defined as equaling seven days. This is what the domain expert indicates is the likely maximum duration between two events (aggressive incidents) where the response of the first incident still has an effect on the behavior of the client in the second incident. Based on , we introduce state . It represents the state where no next incident occurs within the defined duration of . In Table 2 we can see, for example, that distracting the client seems to be related to .

Another step in the pre-processing phase is to check the statistical assumptions. The Chi-square test, which we will elaborate on in the next sections, has a number of assumptions [11]: 1) the data should be frequencies or counts, 2) the categories of the variables are mutually exclusive, 3) each subject may only contribute to one cell (no repeated measures), 4) both variables are measured as categories, 5) the sample size should be sufficiently large. The first four assumptions are data requirements that need to be checked by the analyst that applies the ARE miner. The fifth assumption can be automatically checked by the ARE miner itself.

With regard to the fifth assumption (i.e., sufficient sample size), we implement a heuristic selection criterion: the value of the expected cells in each table should be 5 or greater in at least 80% of the cells [11]. If this criteria is not met, the action-response or response-effects pair is excluded from the analysis and an NA value is the output.

4.2. Stage 1: Computing High-Level Statistics

After pre-processing the event log, we investigate for each action the significant relation between the responses and the effects. In our example, the client shows a certain type of aggressive behavior (the action). Given this, we are interested in how the response of a caretaker to that incident has an impact on the follow-up incident (effect). Hence, we will explain the ARE miner with a fixed initial action. The details are formalized in Algorithm 1. In the following, we will explain how the specific steps from Algorithm 1 are linked to the conceptual considerations.

In Table 3, we show an example of the observed and calculated expected frequencies for the action physical aggression against objects (lines 3 & 4 in Algorithm 1). This allows us to perform a Chi-square test [10] (line 5). Based on a confidence level (usually 95%), the calculated Chi-square (^{2}) test value is compared to the Chi-square distribution to see if there is at least one pair of response-effect significantly different. If the Chi-square test value is insignificant, the action is excluded from the graphical representation (line 21). If the Chi-square is significant (line 6), this indicates that the effects may depend on the response. We then move to the second stage, see Section 4.3.

We demonstrate this first stage of the ARE miner by applying it to a designed example based on our real-world data set presented in Table 3. Based on the observed values, we can calculate the expected values in the table, for example, the expected value for the first cell: response *Terminate Contact* and effects *VA* $= \frac{N_r \times N_c}{E[a][r][c]} = \frac{300 \times 400}{1750} = 68{:}6$. We know from Table 3 that there are five response classes and four effects classes, so the degrees of freedom: $c = (5 \quad 1) \quad (4 \quad 1) = 12$. Given all this, we can calculate the Chi-square test value for the overall table:

$$^2_c = \sum_{i=1}^{} \frac{(O_{\text{Warning;PO}} \quad E_{\text{Warning;PO}})^2}{E_{\text{Warning;PO}}} + \cdots + \frac{(O_{\text{Distract client;}} \quad E_{\text{Distract client;}})^2}{E_{\text{Distract client;}}}$$

$$^2_{12} = \frac{(250 \quad 257{:}1)^2}{257{:}1} + \cdots + \frac{(10 \quad 17{:}7)^2}{17{:}7} = 63{:}47$$

Now we need to determine if this value is significantly different from the mean of the Chi-distribution [12]. The formula for calculating the p-value is complex and will thus not be discussed in detail in this paper. For more details we refer to [12]. In the above example, the p-value ($<$ 0:001) shows that our Chi-square value is significant. This indicates that for at least one pair of response-effects given action PO there is a significant difference from the expected frequency. Thus, we perform a Chi-square test for each individual response.

4.3. Stage 2: Computing Detailed Statistics

In the second stage of the ARE miner, we perform the Chi-square test again on each response class to determine for which response we need to perform post-hoc statistical tests (lines 8 - 10 in Algorithm 1). For this purpose we create dummy variables. A dummy variable is made for each individual response, which takes the value of 0 or 1. The new table we create is a 2 x 4 table where the rows represent the response either taking a 0 or 1 value, see Table 4. Note that the degrees of freedom changes to three. The same formulas are used to calculate the individual response Chi-square score and the corresponding p-value. A Bonferroni correction [13] is made to correct the critical value for the fact that on the same table multiple sets of analyses are performed. The Chi-square test identifies for which responses there is at least one effects that is significantly different from the expected frequency. If the Chi-square value is significant, we create a node for the response and perform post-hoc tests to identify the exact pairs of response-effects that are significant (line 11).

We will demonstrate this stage on an example case. We test five times (one for each response). Thus, we apply the Bonferroni correction [13] on a confidence level of 95% (meaning $= 0{:}05$): $\frac{0:05}{5} = 0{:}01$. If we take Table 4, we can use the same formulas as presented in the previous section to calculate the expected values. Note that we assume independence of responses. Thus, if there are two responses, the action is counted twice: once for response 1 and once for response 2. Therefore, the observed frequencies in Table 3 are not necessarily equal to those in Table 4. If we perform the Chi-square test for the response *Terminate contact* we get a Chi-square score of 31.96 with a p-value $<$ 0:001. Thus, for the response *Terminate contact* there is at least one effects that is significantly different from the expected frequency. In the next section, we describe how a post-hoc test will need to identify the exact pairs for which this is true.

4.4. Stage 3: Identifying Influential Points

In the last stage, the post-hoc tests are performed to test which exact pairs of response-effects have a significant contribution to the Chi-square test value. To do so, the adjusted standardized residuals (ASR) [14] are calculated (line 13 in Algorithm 1). They represent a normalization of the residuals (observed - expected frequency). As the residuals can take either a positive or negative value we use two-sided testing. In order to improve the interpretability, we transform the level into a critical value. We refer to [12] for details on this approach. If $jASRj > criticalscore$ the difference between observed and expected frequency is significant. A significant score means that a specific pair of response-effects has a significant impact on the overall test value. We refer to this as an *influential point*.

For each influential point, arcs are drawn in the graphical representation (lines 14-17). If the score is insignificant, no arc is drawn for that pair of response-effects. We first draw an arc from the action to the responses. Then, we draw an arc from the response to the effect for which we found a significant relation. If the observed frequency is larger than the expected frequency, i.e., the response leads to an increase in frequency of effects, we draw a thick arc. Correspondingly, if the observed frequency is lower than the expected frequency we draw a thin arc. The total number of graphical representations created equals the number of actions for which a significant Chi-square score is found (line 25).

Now, we turn to the example from Table 4. From the previous section we know that the response *Terminate contact* results in a significant Chi-square score. To calculate which points are influential points we calculate the adjusted standardized residuals for each pair. To exemplify, we show the calculation of the ASR for the pair *Terminate contact = 1* and *VA*:

$$ASR = \frac{90 \quad 64{:}3}{64{:}3 \quad (1 \quad \frac{64:3}{300}) \quad (1 \quad \frac{64:3}{300})} = 4{:}08$$

Given our Bonferroni correction gave us an alpha of 0.01 (see previous section), we need to test on the 99 % con dence level. The critical absolute value for this is 2.57. Thus, if our ASR value is> j2:57j, we mark it as an in uential point and draw an arc in the graphical representation. In the example of the pair Terminate contact = 1 and VA the ASR is larger than the critical score (4:08 > 2:57). Therefore, we draw an arc in the graphical representation of this example.

To increase the readability of the graph, we use a variety of arcs. First, on the arc from an action to a response, we indicate the observed frequency of the behavior. This shows how often this speci c action-response pattern is observed. Next, on the arc from a response to an e ects, we display the observed frequency and the expected frequency in brackets. This shows whether or not the response leads to an increase or decrease in the behavior type of the e ects. To also display the strength of the relation between the response and the e ects in the graph, we adjust the thickness of the arc based on the adjusted standardized residuals. Recall that this value needs to be> j2:57j in order for an arc to be drawn. We introduce six classes of e ect strength, i.e. three positive classes and three negative classes. We choose to use a total of six classes to di erentiate between the strength of the e ect as this is a number that is easily comprehensible, yet allows for su cient distinctions between e ect sizes.

We determine the classes by identifying the maximum and minimum ASR scores for each action table. Subsequently, we create the range of ASR scores for each class by dividing the scores between the maximum ASR score and 2.57 equally into three classes. The same structure applies to the negative scores, but then we use the di erence between the minimum ASR score and -2.57. As an example, assume the maximum ASR value is 8.30, and the minimum ASR value is -4.37. The three positive classes will be, from least thick to thickest; (1) [2.57:4.48], (2) [4.49:6.39], and (3) [6.40:8.30]. In line, the three negative classes will be, from least thick to thickest: (1) [-2.57:-3.17], (2) [-3.18:-3.77], and (3) [-3.78:-4.37].

After applying the last stage of the ARE miner on all actions, responses, and e ects on the aforementioned example, we obtain a total of three graphical representations (one for each action). In the next section, we evaluate the ARE miner both on an arti cial as well as a real-world data set to demonstrate that it indeed can generate understandable process representations that provide the user with the required insights into the process execution.

5. Evaluation

The goal of this section is to demonstrate the e ectiveness of the ARE miner to discover models that allow to obtain meaningful insights into action-response-e ect patterns. To this end, we implemented the ARE miner in Python [1] and conducted a quantitative as well as a qualitative evaluation. In the quantitative evaluation (Section 5.1), we use an arti cial data set to systemically explore in a large range of constellations how the representations produced by the ARE miner compare to traditional process-oriented representations. In the qualitative evaluation (Section 5.2), we apply the ARE miner to a real-world action-response-e ect log and investigate to what extent the discovered models are meaningful from a domain perspective.

5.1. Quantitative Evaluation

This section discusses the quantitative evaluation of the ARE miner. Our goal is to develop an understanding of how the representations produced by the ARE miner compare to directly-follows graphs, i.e., traditional process-oriented representations, in a variety of di erent settings. To this end, we generate an arti cial data set that represents a broad spectrum of real-life scenarios. Using this set, we can systematically explore under di ering circumstances how key characteristics, such as the number of arcs, develop. In Sections 5.1.1 and 5.1.2, we rst elaborate on the arti cial data set generation and the setup. Then, in Section 5.1.3, we present the results.

[1] Source code and results: github.com/xxlu/ActionE ectDiscovery

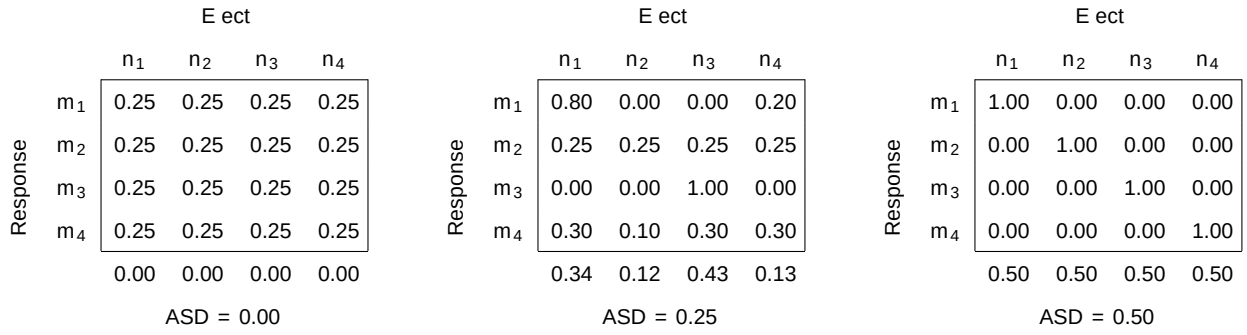

Figure 3: Three possible frequency tables with n=4, m = 4, and Δ = 0.05.

5.1.1. Data Set Generation

To obtain artificial data representing a broad spectrum of possible real-life situations, we generate a set of different frequency tables (see Table 3 for an example). The main rationale behind this approach is that frequency tables summarize all relevant characteristics from the log that we build on in the context of the ARE miner. Hence, generating frequency tables instead of actual action-response-effect logs allows us to precisely control these characteristics.

To illustrate the details of the frequency table generation, recall that a frequency table captures the number of times a response $r \in R$ leads to an effect $e \in E$, given an action $a \in A$. For example, the first row from Table 3 describes how many times we observe a particular effect (i.e., PO, PP, VA, or) for the response Warning. Taking a look at the numbers, we see, for example, that the response Warning leads to PO in 250 cases and to PP in 400 cases. Intuitively, these absolute numbers can also be converted into probabilities. Given the total of 900 observations for the response Warning, we can determine that the probability of a warning leading to PO is approximately 0.28 (250/900). If we determine the probabilities for the other effects as well, we obtain the probability vector (0.28, 0.44, 0.22, 0.06). In statistical terms, this probability vector represents the probability mass function (PMF) of the underlying discrete distribution that we observe in the first row of Table 3. Since such a probability vector can be computed for every row, the frequency table can be described by using m probability vectors $(v_1, \ldots, v_m)$, where m is equal to the number of rows and, therefore, also to the number of responses.

To generate an artificial data set, we build on these probability vectors representing PMFs. The advantage of doing so is that they allow us to systemically capture a large range of possible real-life constellations. Intuitively, there are two extreme scenarios for an action-response-effect pattern. The first is if a considered response is only leading to a single effect as, for instance, described by the probability vector (1.0, 0.0, 0.0, 0.0). The second is if the likelihood for all effects is the same, as, for instance, described by the probability vector (0.25, 0.25, 0.25, 0.25). Besides these two extremes, there is an infinite number of alternative probability vectors for a given a number of n effects. Therefore, we introduce the parameter Δ. By requiring that each probability p_i that is part of a probability vector $v = (p_1, \ldots, p_n)$ is a multiple of Δ, we guarantee that the number of possible probability vectors is finite. Keep in mind that $\sum_{i=1}^{n} p_i = 1$ because we are dealing with vectors representing PMFs. Based on these considerations, we compute the set V of all possible probability vectors for a given number of effects n and the parameter Δ. Given a number of m responses, the total set of possible frequency tables F is then given by the n-ary Cartesian power of V, i.e., $F = V^m = \{(v_1, \ldots, v_m) \mid v_i \in V \text{ for every } i \in \{1, \ldots, m\}\}$. We use $f = (v_1, \ldots, v_m)$ to refer to an individual constellation from F.

To characterize the potentially large number of possible frequency table constellations, we introduce the complexity indicator average standard deviation (ASD). The ASD is the arithmetic mean of the standard deviations of the individual columns of a frequency table $f \in F$. As such, it quantifies to what extent the probabilities of different responses leading to the same effect differ from each other. The closer ASD is to zero, the smaller the differences across the responses. The closer ASD is to the maximum possible average standard deviation for f, the higher the differences across the responses. Note that this maximum value for

ASD depends on the number of responses. If, for instance, $m = 4$, then the maximum ASD is 0.5. Figure 3 shows three possible frequency tables and their ASDs resulting from a generation run with $n=4$, $m = 4$, and $\varepsilon = 0.05$. The left and the right tables show two extremes with the ASD being 0.00 and 0.50. The example in the center shows a rather mixed case. These three examples highlight the broad spectrum of frequency table constellations that may arise in practice and we, therefore, need to systematically consider.

Based on the approach introduced above, we generated an artificial data set with $m = 4$, $n = 3$, $\varepsilon = 0.2$. Note that higher values for parameters m and n as well as a smaller value for ε mainly affect the granularity of the results but not the results themselves. Therefore, we selected parameters that balance granularity and computational effort. In total, this choice of parameters results in a set F containing 175.616 different frequency tables. We will refer to these as constellations. To make sure that the generated constellations also include patterns with a low frequency (i.e., smaller than ε), we randomly add and substract values of 0.01 up to 5 times per row in each $f \in F$. Figure 4 visualizes the distribution of the resulting constellations with respect to the ASD. We can see that it represents a discrete representation of the Bell curve where constellations with an ASD of between 0.20 and 0.25 are most likely.

Figure 4: Distribution of constellations with respect to the ASD.

5.1.2. Setup

In our evaluation experiments, we compare three different techniques:

- ARE miner: We implemented the ARE miner in Python[2] as described in Section 4. Our implementation also automatically performs all assumption checks. Note our implementation may return a graph without any edges in case no significant edges could be identified.
- Naive DFG: As a first baseline, we use a naive directly-follows graph implementation. This configuration returns an arc for every observed response-effect pattern, i.e., for every value in $f \in F$ that is above 0.
- Filtered DFG: As a second baseline, we use a filtered directly-follows graph implementation. This configuration returns an arc for an observed response-effect pattern if the relative frequency of that pattern with respect to the most frequent pattern from the considered constellation $f \in F$ is above the threshold τ. For the purpose of our experiments we set $\tau = 0.8$. This means that if in a constellation f the most frequent pattern occurs with a probability of 0.5, then every pattern with a probability of less than 0.1 will be removed.

For each of the configurations above, we compute which arcs they generate for each $f \in F$. To quantify the results, we determine 1) the number of arcs generated for each f, and 2) the fraction of significant arcs

[2] The code is publicly available for reproducibility: github.com/xxlu/ActionEffectDiscovery.

(a) ARE miner (b) Naive DFG (c) Filtered DFG

Figure 5: Overview of number of generated arcs for each $f \in F$

generated for each f. The first performance measure allows us to understand to what extent the total number of arcs produced by the ARE miner compare to the two baselines. On the one hand, we expect to reduce the overall complexity and, therefore, generate fewer arcs. On the other hand, we want to demonstrate the applicability of the ARE miner, which means that we want to demonstrate that the ARE miner does not generate zero arcs or a single arc in the majority of constellations. The second performance measure helps us to understand how many of the significant arcs generated by the ARE miner are also covered by the two baselines.

5.1.3. Results

Below, we present the results from the quantitative evaluation. We first analyze the number of arcs generated by each technique. Then, we take a detailed look at the fraction of significant arcs.

Number of arcs. The results of our evaluation experiments with respect to the number of arcs are visualized in the bubble charts in Figure 5. The charts show how often a certain number of arcs (y-axis) were generated for a set of constellations from F with a particular ASD (x-axis). A first glance reveals that the representations produced by the ARE miner differ considerably from the respective DFGs. Most notably, on average, the number of arcs generated by the ARE miner is much lower than the number of arcs generated by the naive DFG-based approach. As for the number of arcs, the ARE miner seems to produce, on average, around the same number of arcs as the filtered DFG-based approach. Considering the ASD values, we see that the naive DFG-based approach produces a roughly equal number of arcs over the range of ASD values. If we take a closer look at the ARE miner, we see that it draws less arcs when the ASD is low and more when the ASD is high. We follow up on this observation below. Comparing this to the filtered DFG-based approach, we see that the DFG-based approach seems to do the opposite. It generates more arcs when the ASD is low and fewer when it is high.

Table 5 provides a detailed view on the results. It shows that the average number of arcs produced by the ARE miner in comparison to the naive DFG-based approach is about 4.10 lower (6.83 versus 10.93) and about 0.44 lower than the filtered DFG-based approach (6.83 versus 7.27). Note, however, that this number must be considered in the context of the chosen n and m. Since the maximum number of possible arcs is 12, filtering an average of 4 arcs has a notable effect on the resulting representations. If we take a closer look at the numbers, we can see that, as the ASD increases, the ARE miner draws more arcs and the filtered DFG-based approach draws less. The lower the ASD, the more arcs they contain. This suggest that, while the average number of arcs are almost the same, the filtered DFG-based representations may contain arcs that the ARE miner decided to suppress and tends to miss those the ARE miner decided to draw. This is caused by the fact that the notion of statistical significance is a relative consideration and not based on absolute numbers. Realizing this, the next section looks into more detail into which arcs each technique includes in the respective representations.

Fraction of significant arcs. To understand how the considered configurations differ on a semantic level, it is helpful to analyze which arcs the generated representations have in common. Building on the premise that

		Number of arcs														
	ASD	0	1	2	3	4	5	6	7	8	9	10	11	12	Total	Avg.
ARE miner with noise	0.00-0.05	58	20	36	6	2	0	0	0	0	0	0	0	0	122	0.97
	0.05-0.10	177	417	1129	588	642	179	23	1	0	0	0	0	0	3156	2.55
	0.10-0.15	126	45	459	2145	4829	4680	4577	1194	306	34	1	0	0	18396	4.82
	0.15-0.20	180	0	67	730	3439	8491	12335	11101	5460	1609	262	18	0	43692	6.21
	0.20-0.25	138	0	13	203	1415	5075	11257	15767	13553	6697	2241	318	59	56736	7.14
	0.25-0.30	84	0	8	34	319	1433	4458	8997	10969	7778	3132	688	80	37980	7.79
	0.30-0.35	18	0	0	0	11	137	733	2250	4010	3883	1951	440	43	13476	8.37
	0.35-0.40	0	0	0	0	0	10	44	197	516	644	406	73	0	1890	8.72
	0.40-0.45	0	0	0	0	0	0	6	16	41	65	40	0	0	168	8.70
	Total	781	482	1712	3706	10657	20005	33433	39523	34855	20710	8033	1537	182	175616	6.83
DFG with noise 100%	0.00-0.05	0	0	0	0	0	0	1	0	0	18	18	18	67	122	11.07
	0.05-0.10	0	0	0	0	0	0	0	0	18	108	558	852	1620	3156	11.25
	0.10-0.15	0	0	0	0	0	0	0	15	57	612	2724	6588	8400	18396	11.23
	0.15-0.20	0	0	0	0	0	0	0	21	156	1536	6927	17412	17640	43692	11.16
	0.20-0.25	0	0	0	0	0	0	0	9	288	2454	11631	26340	16014	56736	10.97
	0.25-0.30	0	0	0	0	0	0	0	9	342	3084	11859	16842	5844	37980	10.65
	0.30-0.35	0	0	0	0	0	0	0	0	174	1494	5340	5418	1050	13476	10.42
	0.35-0.40	0	0	0	0	0	0	0	0	42	432	942	456	18	1890	9.99
	0.40-0.45	0	0	0	0	0	0	0	0	6	90	72	0	0	168	9.39
	Total	0	0	0	0	0	0	1	54	1083	9828	40071	73926	50653	175616	10.93
DFG with noise 80%	0.00-0.05	0	0	0	10	0	0	54	9	9	30	3	3	4	122	7.13
	0.05-0.10	0	0	0	96	99	15	252	603	735	546	522	228	60	3156	8.15
	0.10-0.15	0	0	0	24	360	354	792	2865	4791	4938	3357	915	0	18396	8.40
	0.15-0.20	0	0	0	0	534	1245	2181	7440	12855	15255	4176	6	0	43692	8.14
	0.20-0.25	0	0	0	306	1404	2766	5877	15018	24468	6891	6	0	0	56736	7.38
	0.25-0.30	0	0	0	804	3057	5532	7884	14937	5766	0	0	0	0	37980	6.33
	0.30-0.35	0	0	0	354	2946	3498	4158	2520	0	0	0	0	0	13476	5.41
	0.35-0.40	0	0	0	372	870	510	138	0	0	0	0	0	0	1890	4.22
	0.40-0.45	0	0	0	78	90	0	0	0	0	0	0	0	0	168	3.54
	Total	0	0	0	2044	9360	13920	21336	43392	48624	27660	8064	1152	64	175616	7.27

Table 5: Number of arcs generated by each technique

statistically signi cant arcs provide the insights we are looking for from a semantic point of view, we are therefore interested in the fraction of signi cant arcs produced by the ltered DFG-based approach. Note that a comparison with the naive DFG-based approach is obsolete since the naive DFG-based approach will contain all possible and, therefore, also all signi cant arcs. Figure 6 visualizes the number of shared and non-shared arcs for both the ltered DFG-based approach and the ARE miner. More speci cally, it shows how many arcs, on average, are produced for the di erent constellations.

For low values of ASD, we can see that both the number of arcs generated by the ARE miner and the number of shared arcs are very low. However, the number of arcs produced by the ltered DFG-based approach is quite high for low ASD values. Even in the lowest bin, from 0 to 0.05, the ltered DFG-based approach generates an average of 7.1 arcs. With an increasing ASD also the number of shared arcs increases. In general, this is in line with our expectations. The closer we get to an ASD of 0, the more equally distributed is the data. Hence, the ARE miner will identify only a few signi cant arcs, if any. The closer we get to the maximum ASD, the more we face a random distribution. In such a setting, the ARE miner is more likely to identify signi cant arcs and, therefore, generates an increasing number of arcs. Since the number of arcs produced by the ltered DFG-based approach is relatively stable, the number of shared arcs also increases when we move to the high end of the ASD value.

From a semantic perspective, Figure 6 highlights the importance of building on the statistical notion we

Figure 6: Average number of arcs drawn by the ARE miner and the filtered DFG-based approach, plus the number of shared arcs that are drawn when comparing the two techniques.

use in the ARE miner. The filtered DFG-based approach generates a relatively stable number of arcs across all constellations although the number of statistically significant and, therefore, meaningful arcs differs considerably. Constellations with a very low ASD simply do not provide evidence that there are many meaningful patterns to detect. This, however, cannot be captured by filtering arcs based on frequency. It requires the statistical perspective exploited by the ARE miner.

In summary, the quantitative evaluation illustrates that the ARE miner performs well in a broad range of possible situations. We showed that 1) the ARE miner leads to a notable reduction in the number of arcs compared to the naive DFG-based approach, and 2) the ARE miner produces a different, and more meaningful, set of arcs than the filtered DFG-based approach. This highlights the value of building on the notion of statistical significance in this setting. Next, it is interesting to apply the ARE miner on a case study to investigate if the graphs produced by the ARE miner can provide relevant and meaningful domain-specific insights.

5.2. Qualitative Evaluation

This section discusses the qualitative evaluation the ARE miner. Our goal is to demonstrate the effectiveness of the ARE miner to discover models that allow to obtain meaningful insights into action-response-effect patterns.

5.2.1. Data set

To evaluate the ARE miner, we use a real-world data set related to the care process of a Dutch residential care facility. The event log contains 21,384 recordings of aggressive incidents from 1,115 clients. The process captured in this log concerns the aggressive behavior of clients in their facilities and the way client caretakers respond to these incidents. The log consists of aggressive incidents of clients that belong to one of four different action classes. Each of these actions is followed by a number of measures from the caretakers as responses to the action. Each response belongs to one of nine different response classes. In line with the

Actions	Physical aggression towards people	11,381
	Physical aggression towards objects	1,446
	Verbal aggression	5,778
	Self-injury	2,779
	Total	21,384
Responses	Talk to client	9,279
	Held with force	3,624
	Leave room	3,638
	Distract client	2,561
	Send away	3,169
	Seclusion	1,156
	Other measures	209
	None	783
	Ignore client	70
	Total	24,489
E ects	Physical aggression towards people	5,897
	Physical aggression towards objects	686
	Verbal aggression	2,369
	Self-injury	1,429
	No next incident ( )	9,888
	Total	20,269
Clients	Minimum number of actions per client	1
	Maximum number of actions per client	449
	Average number of actions per client	19.2
	Total	1,115

Table 6: Overview of the characteristics of the real-world data set

description of the ARE miner, we transformed this log into an action-response-e ect log by de ning the next aggressive incident of a client as e ect, given it occurred within an of 9 days as indicated based on the data. Otherwise, the e ect is determined with . As a result, we obtain a total of ve di erent e ect classes. Table 6 summarizes the characteristics of our data set.

5.2.2. Results

Below we present the results from the qualitative evaluation. We focus on three particular aspects: 1) the interpretation of the resulting graphs, 2) the insights we can obtain from these graphs, and 3) how the graphs compare to directly-follows graphs.

Interpretation. After applying the ARE miner to the data set, we obtain four graphs, one for each action class. In Figure 7 we show the resulting graphs for each action. Each arc in these graphs denotes an in uential point representing a response-e ect interaction. Recall that the number of observed instances for in uential points is signi cantly higher (solid arc) or lower (dotted arc) than the statistically expected number of instances. In addition, the thickness of the arc visualizes the size of the e ect, i.e., the thicker the arc, the stronger the e ect. As illustrated by the graphs, the resulting number of arcs is, despite the complexity of the log, quite low since only a few arcs represent statistically signi cant interactions. This allows us to study the impact of each response to an action in detail.

To illustrate this, consider Figure 7a, where the initial action is Physical aggression against objects (po_s). Among others, this graph reveals that Terminate contact has been observed 299 times in our data set as a response to Physical aggression against objects. We can further see that in 48 instances, this response has led to the e ect Verbal aggression. In brackets we can see that the expected value based on statistics for this arc is 32 instances. Thus, there are signi cantly more instances of Verbal aggression after a Terminate contact response than statistically expected, which is visualized using a solid arc. If we consider the response Seclusion, we can see that we observe an opposite e ect for Tau ( ). Here, the observed

number of instances (27) is signi cantly lower than the expected number of instances (45). Therefore, the interaction is visualized using a dotted arc. What both cases have in common is that they represent statistically signi cant interactions, which increase our understanding of what likely our unlikely e ects a particular response will trigger.

Insights. There are a number of relevant insights we can obtain from the graphical representations in Figure 7. For instance, Figure 7a reveals that Seclusion is not a good response to the action Physical aggression against objects since it results in a signi cantly higher number of instances of Physical aggression against people (PP). The frequencies show that the response Seclusion is almost 1.7 times as likely to result in e ect Physical aggression against people than statistically expected. At the same time, the response Seclusion leads to a signi cantly lower likelihood of having no further aggression incident, as indicated by Tau. This highlights even further that Seclusion as a response to Physical aggression against objects is not a preferable choice.

In a similar fashion, we can interpret the resulting graphs for the other three actions. However, particularly Figure 7c and Figure 7d highlight that also focusing on statistically signi cant interactions may result in relatively complex representations. From an insight perspective, we can make three main observations: 1) depending on the action, the same response may lead to di erent outcomes, 2) some response-e ect pairs are only signi cant for some actions, and 3) some response-e ect pairs are signi cant across all actions.

The rst observation is best illustrated by the response No measures. We can see that the response No measures leads to very di erent outcomes in Figure 7b and Figure 7c. If No measures is used as a response to the action Verbal aggression (Figure 7b), we observe fewer instances of Tau than statistically expected. In other words, it leads to an escalation of the aggression since it is less likely that no next incident occurs. By contrast, if No measures is used as a response to the action Self-injurious behavior (Figure 7c), it leads to a signi cantly higher number of instances of Tau than statistically expected. What is more, it leads to signi cantly fewer instances of Physical aggression towards people than statistically expected. Both these outcomes can be considered as a deescalation of the aggressive behavior which are valuable insights for the management of this behavior.

For the sake of illustrating the second observation, consider the response Talk to client, which only occurs as part of a signi cant interaction in the context of the action Self-injurious behavior (Figure 7c). Here, we see that it leads to an escalation of violence, i.e., to signi cantly more Physical aggression towards people.

An example for the third observation is the response Seclusion, which has a similar e ect across all four actions. We see that it leads to signi cantly more Physical aggression towards people and signi cantly less to Tau. This means that we see more of the most severe form of aggressive behavior and, at the same time, a lower likelihood of no next aggressive incident. Thus, we can globally speak of an escalation of the violence as a result of this response. Notice that for Self-injurious behavior there is no e ect. This is logical considering that this response is used to restrain a client from being violent. As such, this response is rarely used in a setting where the victim is the client him/herself.

Comparison to directly-follows graph. Figure 8 shows the directly-follows graphs obtained for the actions Physical aggression towards objects and Physical aggression towards people using the process mining tool Disco. For the sake of readability, the ltering settings are set to 5% and 1% respectively, i.e., only the 5% and 1% most frequent action-response-e ect patterns are included. A brief analysis of the graph reveals that it does not allow us to obtain the same insights as the miner proposed in this paper. Most notably, the process model contains a large number of arcs. Given that our data set contains four action classes, nine response classes, and ve e ect classes, the directly-follows graph can potentially contain 81 ($9 \cdot 4 + 9 \cdot 5$) arcs. Already a single instance of a particular response-e ect pattern for a considered action will result in additional arc. The number of arcs increase exponentially with the number of responses observed. A possible solution to this could be to add information to the control- ow based representation, such as the observed frequencies of the arcs or nodes.

However, ltering based on the frequencies does not always deliver the desired result. This is also illustrated in Figure 8. It shows that ltering could even be misleading since the data set is imbalanced. In this real-world scenario, a high frequency does not imply a signi cant pattern. This becomes obvious if

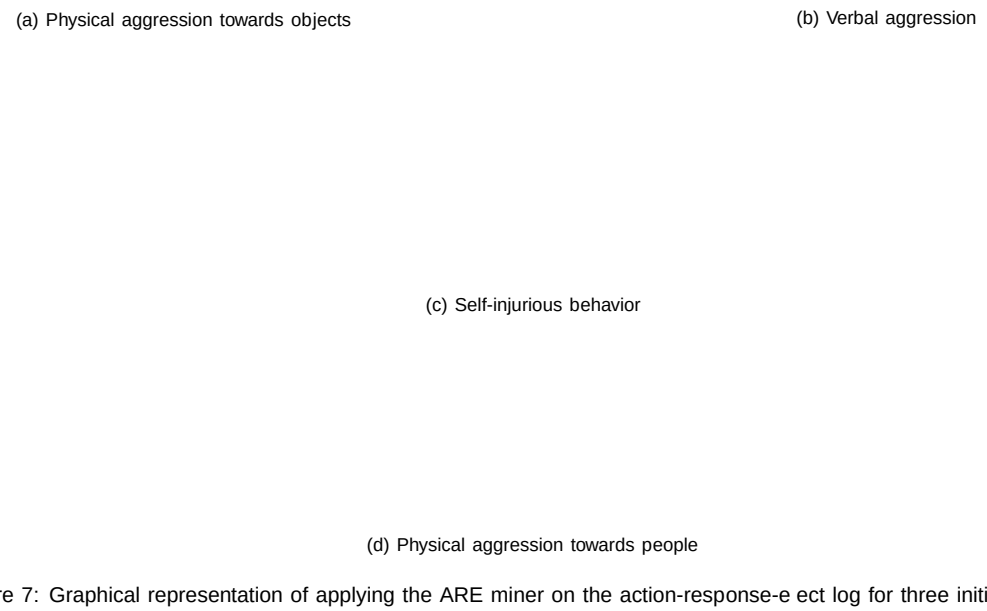

Figure 7: Graphical representation of applying the ARE miner on the action-response-e ect log for three initial actions.

we compare the techniques. From the gures we can see that none of the signi cant response-e ect pairs from Figure 7a are displayed in Figure 8a. In addition, for the most complex action, only one pattern (PP - Distract client - tau/) of response-e ect can be observed in Figure 8b.

In order to understand the relations in the representation, we have to account for the relative frequencies. These reveal the meaningful insights that are hidden in the representation of a discovery technique such as the directly-follows approach. Hence, even after applying ltering mechanisms, Figure 8 does not provide the necessary insights. For example, even though we can observe a pattern for the actionPhysical aggression towards people (PP) in both our graph and the directly-follows graph, we cannot determine the meaning of the arc in the latter. We cannot assess whether the frequency that is displayed on the arc (22) is a statistically relevant e ect. Hence, the directly-follows graph does not provide the insights that are required to answer question such as: If a client displays aggressive behavior of class X, which response is likely to lead to an (de-)escalation or future aggression? If we consider the same pattern again for the actionPP, we can see in our graphical representation that we show that the responseDistract client leads to signi cantly fewer instances ofTau. This means that this particular response to the considered action seems to escalate violence, after all the chance of no next incident occurring is lower than we would expect based on statistics.

The qualitative evaluation in this section using a real-world data set strongly suggests that the representations generated by the ARE miner allow to obtain relevant domain-speci c insights that can be directly translated into practical guidance.

6. Discussion

In this section, we discuss the implications as well as the limitations of the work presented in this paper.

Implications. The key question identi ed at the start of this research addressed the desire to express insights into how a response to an action can lead to a desired or undesired outcome (e ect). In our problem

(a) Physical aggression towards objects

(b) Physical aggression towards people

Figure 8: Directly-follows process model of the real-world event log for the initial action physical aggression against objects (PO) and physical aggression towards people (PP). This shows the process filtered on 5% of the possible activities and paths for the initial action PO and 1% of the possible activities and paths for the initial action PP. The models were created using Disco [3].

statement we identified two main challenges associated with this that need to be overcome: (1) graphical representation, and (2) effective filtering mechanism. Our evaluation uses an artificial log covering a broad range of scenarios that highlights how the proposed ARE miner addresses both these challenges. In addition, we evaluate the ARE miner a real-world data set from the healthcare domain. Figure 7 shows that the ARE miner creates a simple graphical representation that allows for insights into statistical relations that cannot be obtained using Figure 8. In addition, we show that the use of statistics substantially reduces the number of arcs in comparison to a naive DFG-based approach. The filtering mechanism is also effective in the sense that it filters those arcs that are meaningful, opposed to those that are merely frequent.

One interesting implication of the ARE miner generated insights can be used to support decision-making processes. In our example, Figure 7 can be used to train existing and new staff members to ensure that appropriate responses are taken. For example, one could show that responding with seclusion will likely escalate future violent behavior of the client. Placing the ARE miner in a broader medical context, it could help make informed decisions when different treatment options are considered. In a different domain, the ARE miner could help a marketing organization understand the effectiveness of marketing strategies in terms of response of potential customers. In short, the ARE miner provides insights into action-response-effect patterns where the objective of analyzing the process is to understand possible underlying statistical dependency patterns.

Limitations. The work presented in this paper is subject to a number of limitations, which relate to the ARE miner itself as well as the experimental evaluation.

As for the ARE miner, there are three main limitations. First, we assume the independence of the responses. This means that each response has a unique impact on the effect and there is no interaction effect when responses are combined. For example, if response r_1 is observed to lead to effect c_1 and response r_2 is observed to lead to effect c_2, then only these independent patterns will be included even if the combination of r_1 and r_2 actually leads to c_3. Adjusting for this, would require a new formalization and introduce considerable additional complexity since the set of responses would be no longer R, but $R \subseteq R$. Second, our formalization defines an effect as the next occurrence of an action after the response. In certain scenarios, it could be very interesting to consider the generalize this formalization by allowing the effect to be any type of event or activity. However, with such a generalization we can no longer compare the ARE miner with a directly-follows graph. As such, such a fundamental adjustment requires an entirely different means of comparison, which is left for future research. Third, we need to consider the data requirements of the ARE miner. The scenario in which the ARE miner is mostly applicable is when there is a choice to be made in the process. Hence, a form of categorical data needs to be available. In addition, the ARE miner does not allow for continuous variables to be included. However, continuous variables can often be transformed to

categorical variables.

The main limitation that needs to be considered for the experimental evaluation concerns the parameter settings for the arti cial data. There are a variety of parameter settings that could be further explored. For example, changing the value of in the PMF generation phase from 0.2 to 0.1 would generate more precise insights into the mechanisms underlying the techniques. In addition, varying the number of responses and e ects may lead to further interesting insights. There are two main reason why we did not address these limitations in this paper. First, the adaptation of these parameters cannot be expected to provide fundamentally di erent insights since they will mainly a ect granularity and size of the data. Second, the experiments for the presented setting already required substantial computing power. Therefore, we decided to stick to the chosen setting.

A limitation that is more of secondary nature is the variety of ways in which the frequency lter for the DFG-based approach can be implemented. Most lters aim to capture one of two key elements of the DFG-based approach: time or activities/paths. The ARE miner focuses on the ltering of activities (nodes) and paths (edges). Time, in the ARE miner, is represented via the de nition of and thus represents a constant. Therefore, ltering based on the frequency of paths provides results that are best suitable for comparison.

7. Related Work

Over the last two decades a plethora of process discovery techniques have been proposed [15]. The majority of these approaches generate procedural models such as Petri nets [16, 17], causal nets [18, 19], BPMN models [20, 21] or process trees [22, 7]. Some techniques also discover declarative models [23, 24] or hybrid models (i.e. a combination of procedural and declarative models) [25, 26]. What all these techniques have in common is that they aim to discover the control ow of a business process, that is, the execution constraints among the process' activities. The ARE miner clearly di ers from these traditional process discovery techniques by focusing on action-response-e ect patterns instead of the general control ow.

There are, however, also alternative approaches to process discovery. We distinguish two prominent classes of techniques: artifact-centric process discovery and causal mechanism discovery. Several authors addressed the problem of artifact-centric process discovery [27, 28, 29]. The core idea of artifact-centric process discovery is to consider a process as a set of interacting artifacts that evolve throughout process execution. The goal of artifact-centric discovery, therefore, is to discover the lifecyles associated with these artifacts and the respective interactions among them. While artifact-centric discovery techniques move away from solely considering the control- ow of the process activities, the main goal is still control- ow oriented. A related technique to process discovery was proposed in [30, 31]. This technique focuses on the di erent perspectives of a process and discovers and captures how their relations change using composite state machines. While the techniques from [30, 31] are potentially useful in many scenarios we address with the ARE miner, the insights that can be obtained with the ARE miner di er substantially. The techniques from [30, 31] allows to understand how di erent artifact life cycle states are related. For example, it reveals that a patient in the state \ Healthy" does no longer require a \Lab test". The goal of the ARE miner is to show what actually needs to be done (or should not be done) to make sure a patient ends up in the state \ Healthy".

The second, prominent set of discovery techniques study the phenomenon of causality in process mining [32, 33]. In [32], the authors investigate how a treatment can have a (high) causal outcome for certain subgroups of cases. In their work, they propose an action-rule based technique in which uplift trees are used to determine the subgroups for which the causal relations are relevant. In [33], the authors rather look at the context in which the process takes place and run a Dynamic Bayesian Network model to determine causal relations. When we compare the ARE miner to previous work, we see that there are three main di erences: (1) the de nition of subgroups, (2) the transparency of the technique, and (3) comprehensibility of the output. We discuss each of these di erences in detail below.

First, the techniques di er in the way subgroups are de ned. One strand of literature, the rule-based approaches and related works, allow for the discovery of subgroups based on data. The main advantage of

the data discovery is that new subgroups can be discovered. By contrast, other techniques take subgroups that are de ned by the user as an input. The main advantage of the user-de ned subgroups is that the subgroups intuitively make sense to the user of the approach. Therefore, the results of this user-de ned approach provides results that are inherently meaningful and actionable to the user.

Second, the transparency of these techniques di ers greatly. Previous research has shown that this plays a crucial role in the acceptance of and trust in new techniques. In [34], the authors showed the crucial importance of understanding how a prediction is made for people in the medical domain in order to come to a decision about a patient's treatment. In [35], the authors expand on this by showing the same holds for other business domains. The main advantage that the machine learning-based techniques have is that they are very powerful and can result in highly accurate results. However, the transparency of the process of obtaining the results is a known dilemma in machine learning techniques [36]. As a result, the movement of explainable arti cial intelligence prescribes that retrospectively additional models can be built to gain insights into this process. According to [37], we can distinguish two counteractions to this: explainable by design or explainable post-hoc. The latter is used in the works of both [32] (uplift trees) and [33] (sensitivity analysis). Using a statistical approach, like we do in this paper, the transparency of the technique is provided by design. All outcomes of the technique are traceable and can be recalculated manually. The main advantage of this is that the ARE miner is intrinsically transparent and thus ensures insights into to why and where questions, e.g., why do certain (causal) relations hold and where they do these relations origin?

Lastly, the output of the approaches di ers substantially. For rule-based approaches the output is declarative in the sense that a set of textual rules are de ned to which a case should hold in order to optimize the e ect. In addition, in [33], the authors provide the user with probabilistic parameters as output variables. The ARE miner also produces probabilistic parameters, but not as output. The ARE miner builds on the parameters by introducing an extra translation of the results into graphical representations with e ect size indications. In this way, we provide the user with an understandable and actionable representation.

To the best of our knowledge, we are the rst to propose a technique, the ARE miner, that discovers action-response-e ect patterns and allows the reader to develop an understanding of why certain events occur. The ARE miner creates this understanding by having user-de ned subgroups, which are used in a transparent technique to produce probabilistic visual output that is intuitive for the user.

8. Conclusions

This paper presented the ARE miner to discover action-response-e ect patterns within work processes. We identi ed two main challenges that we addressed in this research: (1) comprehensible graphical representation, and (2) e ective ltering mechanism. In order to address these challenges, we proposed the ARE miner that builds on ltering in uential relations using statistical tests. We evaluated the ARE miner in two ways. First, we used an arti cial data set to compare the performance of the ARE miner to traditional process-oriented representations. The results show that the ARE miner leads to both: (1) a reduction in the number of arcs drawn, and (2) a set of arcs that is di erent and more meaningful compared to the DFG-based approaches. Second, we evaluated the ARE miner on a real-world data set from the healthcare domain. More speci cally, we used the ARE miner to study aggressive behavior and show that we can gain valuable and novel insights from the representations discovered by the ARE miner. The representations show that the ARE miner can tackle both challenges by providing an easy-to-interpret representation that only displays meaningful relations such that it highlights informative insights.

In future work, we plan undertake a number of steps to extend this work. In line with the limitations we presented previously on the ARE miner, we plan to conceptually extend this work by 1) developing an extension to the ARE miner that can estimate and incorporate the interaction e ect that can arise when there are multiple responses to an action and 2) revisiting the concept of e ects to see if we can relax the formalization to allow for di erent types of e ects other than the next action of a process. Besides these conceptual extensions, we also plan to conduct additional evaluations. Most importantly, we intend to further test the ARE miner on additional real-world cases. What is more, we plan to compare the results of

the ARE miner to machine learning-based approaches. In this way, we can obtain further insights into the applicability and the value of the ARE miner.

Acknowledgments This research was supported by the NWO TACTICS project (628.011.004) and Lunet Zorg in the Netherlands.

References

[1] J. De Weerdt, A. Schupp, A. Vanderloock, B. Baesens, Process mining for the multi-faceted analysis of business processes—a case study in a financial services organization, Computers in Industry 64 (1) (2013) 57–67.

[2] E. Rojas, J. Munoz-Gama, M. Sepúlveda, D. Capurro, Process mining in healthcare: A literature review, Journal of Biomedical Informatics 61 (2016) 224 – 236.

[3] W. M. P. Van der Aalst, Data science in action, in: Process Mining, Springer, 2016, pp. 3–23.

[4] M. Thiede, D. Fuerstenau, A. P. B. Barquet, How is process mining technology used by organizations? a systematic literature review of empirical studies, Business Process Management Journal (2018).

[5] A. J. Weijters, W. M. P. Van der Aalst, Rediscovering workflow models from event-based data using little thumb, Integrated Computer-Aided Engineering 10 (2) (2003) 151–162.

[6] C. W. Günther, W. M. P. Van Der Aalst, Fuzzy mining–adaptive process simplification based on multi-perspective metrics, in: International conference on business process management, Springer, 2007, pp. 328–343.

[7] S. J. Leemans, D. Fahland, W. M. P. van der Aalst, Discovering block-structured process models from event logs-a constructive approach, in: International conference on applications and theory of Petri nets and concurrency, Springer, 2013, pp. 311–329.

[8] J. J. Koorn, X. Lu, H. Leopold, H. A. Reijers, Looking for meaning: Discovering action-response-effect patterns in business processes, in: International Conference on Business Process Management, Springer, 2020, pp. 167–183.

[9] W. M. van der Aalst, A practitioner's guide to process mining: Limitations of the directly-follows graph, Procedia Computer Science 164 (2019) 321 – 328.

[10] W. G. Cochran, The 2 test of goodness of fit, The Annals of Mathematical Statistics (1952) 315–345.

[11] M. L. McHugh, The chi-square test of independence, Biochemia medica: Biochemia medica 23 (2) (2013) 143–149.

[12] R. A. Fisher, F. Yates, Statistical tables: For biological, agricultural and medical research, Oliver and Boyd, 1938.

[13] W. Haynes, Bonferroni Correction, Springer New York, New York, NY, 2013, pp. 154–154.

[14] A. Agresti, Categorical data analysis, Vol. 482, John Wiley & Sons, 2003.

[15] A. Augusto, R. Conforti, M. Dumas, M. La Rosa, F. M. Maggi, A. Marrella, M. Mecella, A. Soo, Automated discovery of process models from event logs: Review and benchmark, IEEE TKDE 31 (4) (2018) 686–705.

[16] W. Song, H.-A. Jacobsen, C. Ye, X. Ma, Process discovery from dependence-complete event logs, IEEE Transactions on Services Computing 9 (5) (2015) 714–727.

[17] H. Verbeek, W. M. P. van der Aalst, J. Munoz-Gama, Divide and conquer: A tool framework for supporting decomposed discovery in process mining, The Computer Journal 60 (11) (2017) 1649–1674.

[18] H. Nguyen, M. Dumas, A. H. ter Hofstede, M. La Rosa, F. M. Maggi, Mining business process stages from event logs, in: International Conference on Advanced Information Systems Engineering, Springer, 2017, pp. 577–594.

[19] B. N. Yahya, M. Song, H. Bae, S.-o. Sul, J.-Z. Wu, Domain-driven actionable process model discovery, Computers & Industrial Engineering 99 (2016) 382–400.

[20] A. Augusto, R. Conforti, M. Dumas, M. La Rosa, Split miner: Discovering accurate and simple business process models from event logs, in: 2017 IEEE International Conference on Data Mining (ICDM), IEEE, 2017, pp. 1–10.

[21] S. K. vanden Broucke, J. De Weerdt, Fodina: A robust and flexible heuristic process discovery technique, decision support systems 100 (2017) 109–118.

[22] J. C. Buijs, B. F. van Dongen, W. M. P. van der Aalst, A genetic algorithm for discovering process trees, in: 2012 IEEE Congress on Evolutionary Computation, IEEE, 2012, pp. 1–8.

[23] M. L. Bernardi, M. Cimitile, C. Di Francescomarino, F. M. Maggi, Using discriminative rule mining to discover declarative process models with non-atomic activities, in: International Symposium on Rules and Rule Markup Languages for the Semantic Web, Springer, 2014, pp. 281–295.

[24] S. Schönig, A. Rogge-Solti, C. Cabanillas, S. Jablonski, J. Mendling, Efficient and customisable declarative process mining with sql, in: International Conference on Advanced Information Systems Engineering, Springer, 2016, pp. 290–305.

[25] J. De Smedt, J. De Weerdt, J. Vanthienen, Fusion miner: Process discovery for mixed-paradigm models, Decision Support Systems 77 (2015) 123–136.

[26] F. M. Maggi, T. Slaats, H. A. Reijers, The automated discovery of hybrid processes, in: International Conference on Business Process Management, Springer, 2014, pp. 392–399.

[27] X. Lu, M. Nagelkerke, D. van de Wiel, D. Fahland, Discovering interacting artifacts from erp systems, IEEE Transactions on Services Computing 8 (6) (2015) 861–873.

[28] E. H. Nooijen, B. F. van Dongen, D. Fahland, Automatic discovery of data-centric and artifact-centric processes, in: International Conference on Business Process Management, Springer, 2012, pp. 316–327.

[29] V. Popova, D. Fahland, M. Dumas, Artifact lifecycle discovery, International Journal of Cooperative Information Systems 24 (01) (2015) 1550001.

[30] M. L. van Eck, N. Sidorova, W. M. P. van der Aalst, Discovering and exploring state-based models for multi-perspective processes, in: International Conference on Business Process Management, Springer, 2016, pp. 142–157.

[31] M. L. van Eck, N. Sidorova, W. M. P. van der Aalst, Guided interaction exploration in artifact-centric process models, in: 2017 IEEE 19th Conference on Business Informatics (CBI), Vol. 1, IEEE, 2017, pp. 109–118.

[32] Z. D. Bozorgi, I. Teinemaa, M. Dumas, M. La Rosa, A. Polyvyanyy, Process mining meets causal machine learning: Discovering causal rules from event logs, in: 2020 2nd International Conference on Process Mining (ICPM), IEEE, 2020, pp. 129–136.

[33] J. Brunk, M. Stierle, L. Papke, K. Revoredo, M. Matzner, J. Becker, Cause vs. effect in context-sensitive prediction of business process instances, Information Systems 95 (2020) 101635.

[34] E. Shortliffe, Computer-based medical consultations: MYCIN, Vol. 2, Elsevier, 2012.

[35] D. Martens, F. Provost, J. Clark, E. J. de Fortuny, Mining massive fine-grained behavior data to improve predictive analytics., MIS Quarterly 40 (4) (2016).

[36] A. Adadi, M. Berrada, Peeking inside the black-box: A survey on explainable artificial intelligence (xai), IEEE Access 6 (2018) 52138–52160.

[37] M. Du, N. Liu, X. Hu, Techniques for interpretable machine learning, Communications of the ACM 63 (1) (2019) 68–77.