# Root Cause Analysis in Process Mining with Probabilistic Temporal Logic

Greg Van Houdt[1(✉)] , Benoît Depaire[1] , and Niels Martin[1,2]

[1] Research Group Business Informatics, UHasselt - Hasselt University, Hasselt, Belgium
{greg.vanhoudt,benoit.depaire,niels.martin}@uhasselt.be
[2] Research Foundation Flanders (FWO), Brussels, Belgium

**Abstract.** Process mining is a research domain that enables businesses to analyse and improve their processes by extracting insights from event logs. While determining the root causes of, for example, a negative case outcome can provide valuable insights for business users, only limited research has been conducted to uncover true causal relations within the process mining field. Therefore, this paper proposes AITIA-PM, a novel technique to measure cause-effect relations in event logs based on causality theory. The AITIA-PM algorithm employs probabilistic temporal logic to formally yet flexibly define hypotheses and then automatically tests them for causal relations from data. We demonstrate this by applying AITIA-PM on a real-life dataset. The case study shows that, after a well-thought-out hypotheses definition and information extraction, the AITIA-PM algorithm can be applied on rich event logs, expanding the possibilities of meaningful root cause analysis in a process mining context.

**Keywords:** Process Mining · Root Cause Analysis · Probabilistic Temporal Logic · Event Log

## 1 Introduction

Process mining is a research domain that enables businesses to analyse and improve their processes by extracting insights from event logs [1]. The foundation is the event log, which records the real execution of a business process. It can then be used for, among other goals, process discovery [2] and conformance checking [6]. However, merely discovering how a process is actually executed and where it differs from the normative model might not be sufficient. Insights in, for example, why an event was triggered or why a trace ended with an exception can be of more interest to business users, and thus, accurate root cause analyses (RCA) are desired.

Identifying root causes can be a complex task [17]. Each process involves many different steps, and for each step many factors can be of influence. Add to this that many traces in a business process can show unique behaviour, as well as

influence each other by having to share resources. Previous research has proposed techniques to conduct RCA in process mining, e.g. [7,10,11], however, there are clear limitations. First, they often put forward a correlation analysis instead of a true RCA. However, when a process characteristic is correlated with a particular undesirable outcome, this does not imply that this characteristic caused the phenomenon. In that sense, one must acknowledge confounding factors can exist, which might cause spurious associations to arise [24]. Second, existing RCA techniques that build upon causality theory impose heavy assumptions on the underlying data. Think of only being able to handle linear causal relations, for example.

Against this background, this paper proposes the AITIA-PM algorithm. This algorithm is a new way of executing an RCA in process mining, inspired by the work of Kleinberg [13,14]. Not only is AITIA-PM based on causality theory, this technique does not impose assumptions on the required data, making it more reliable in the real world. We propose the use of probabilistic temporal logic (PTL) to formally define hypotheses about causal relations, which offers great flexibility. Additionally, we explicitly take confounding factors into account. As such, AITIA-PM is a new addition to the current state-of-the-art of meaningful RCA in process mining. Our contributions are best summarised as follows:

– We propose a novel method in AITIA-PM, adding a new technique to the mix for effective root cause analysis in the process mining domain which is fully based on existing causality theory.
– The demonstration on a real-life event log shows the value of AITIA-PM, mainly found in the flexibility of PTL when identifying specific causal relations and how statistical significance can be computed. It also shows the importance of a theoretical foundation regarding the philosophy surrounding causality, as results are easy to interpret.

The remainder of this paper is structured as follows. Section 2 describes the related work in root cause analysis from a process mining standpoint, after which Sect. 3 introduces the AITIA-PM algorithm which is employed in the demonstration as discussed in Sect. 4. Finally, we conclude our paper in Sect. 5.

## 2    Related Work

An RCA is not bound to a specific family of techniques. Examples are (i) classification techniques as seen in, for example, [3,8,10,22,23], and (ii) rule mining algorithms like association rules [5] and subgroup discovery [19]. Unfortunately, in most applications, there is too little attention given towards the differentiation between correlation and causality.

Hompes et al. [11] proposed a graph-based approach resulting in a time series analysis to detect cause-effect relations by testing for Granger causality [9], thus explicitly considering causation instead of correlation between features. However, it is not perfect either. Granger causality, as it is originally defined, cannot account for instantaneous or nonlinear causal relations, and cannot deal with

confounding effects either. Also, Granger causality makes strong assumptions on the underlying data which are rarely met in the real world [15].

Finally, Qafari and van der Aalst have recently published research on structural equation models for RCA [17] which was later extended with counterfactual reasoning [18]. One of the foundations here is that the structure of causal relations can be provided by the domain expert if available and, as such, there can be no discussion about causality or correlation. The counterfactual reasoning extension allows the authors to produce recommendations that indicate how specific cases could have been handled differently to avoid problems in the future [18]. However, the authors acknowledge that using a machine learning technique imposes the risk of obtaining wrong or imprecise recommendations, or even miss out on the correct ones, regardless of the model's accuracy. Narendra et al. [16] also show how to answer the what-if questions via structural causal models and counterfactual reasoning, proving the effectiveness of the methods, yet they acknowledge it lacks intuitiveness.

The causality measure and complementary algorithm introduced by Kleinberg [13,14] pays great attention towards determining causality by building on the philosophical foundations of causality theory [12,21]. To that end, the algorithm is able to detect the genuine causal relations from data separate from spurious ones. This is achieved by implementing probabilistic temporal logic (PTL) for defining hypotheses, which are then tested based on probability theory and statistical significance. Additionally, Kleinberg's technique explicitly tackles confounding variables.

## 3   The AITIA-PM Algorithm

As described in Sect. 2, Kleinberg's work found its basis in causality theory. The measure and complementary algorithm allow for extraction of causal relations from data rather than a predefined model of how a system evolves in terms of states it is in. AITIA-PM tailors the ideas of Kleinberg to the process mining field. The following paragraphs describe the necessary background followed by a step-by-step guide of the algorithm. For more information, we refer the reader to Kleinberg [13].

### 3.1   Background

**The Concept of Causality.** In this paper, consistent with the work of Kleinberg [13], the following properties must hold to establish a causal relationship between a cause and an effect: (i) the cause must precede the effect in time [12] and (ii) a cause must raise the probability of the effect [21]. Property (ii) is also known as the prima facie condition. Several pitfalls must be taken into account, however.

First of all, there might be causality without raising the probability of the effect or vice versa. For example, yellow stained fingers and lung cancer can be the result of a common earlier cause: smoking. Without considering smoking, one

would observe that having yellow stained fingers would increase the probability of lung cancer. However, when holding the common cause fixed, that relationship between the effects would disappear. Controlling for common causes is known as screening off, or dealing with confounding factors [24].

Second, event logs carry a case notion. However, process instances can influence each other. Think of resources being shared or scarce materials suddenly becoming unavailable because the last item was just consumed, thus impacting how a different case can continue. Therefore, we add another property to AITIA-PM one must meet, namely that (iii) each case is defined by the events which can possibly be a cause of the effect within that specific case.

Clearly, unlike the heavy assumptions made in Granger causality which are, among others, that there is no confounding variable present, causal relations are linear and time series are stationary [15], our understanding of causality imposes less restrictions on the input data. The first two properties, as will be made clear in the following subsections, are also easy to infer from an event log automatically, making inference practically feasible as well.

**Probabilistic Temporal Logic.** PTL allows reasoning on the likelihood of an event within a certain time interval. For example: how likely is it that a train arrives at the station within 2 to 10 min. As such, properties should not hold *eventually*, as they are bound in time so it can be quantified how likely it will happen. By allowing to freely define the cause, effect, type of relation between cause and effect, and the time window, PTL is highly flexible in execution.

AITIA-PM uses PTL as language to define the hypotheses the business user desires to test for cause-effect relations. Each hypothesis comprises a logical formula describing both the time bounds as well as the likelihood of a potential cause $c$ triggering an effect $e$: $c \rightsquigarrow_{\geq p}^{\geq r, \leq s} e$. This is also called a leads-to formula where $r$, $s$ represent the time bounds and $p$ the minimum probability for the cause triggering the effect in the time window in order for the formula to evaluate to true. $c$ and $e$ here are state formulas: properties which hold for the system at a certain point in time. Such a property can be an activity that was executed. For example, with $\neg H$ and $F$ being not doing homework and failing a test respectively, $\neg H \rightsquigarrow_{\geq 0.40}^{\geq 1, \leq 3} F$ would describe that when a student neglects the necessary homework, the probability of the student failing a test between 1 and 3 time units would be at least 40%. From the practical viewpoint of AITIA-PM, the probabilities are calculated from data and do not need to be passed by the user.

The state formulas for the cause and effect are not limited to contain one element each. PTL allows for each state formula to be a path formula too. A path formula can express properties along a path (or trace) in the dataset. For example, a path formula can be that an activity B must follow activity A in a trace within 5 time units, like so:

$$[AF_{\geq p_1}^{\leq 5} B] \rightsquigarrow_{\geq p_2}^{\geq r, \leq s} e \tag{1}$$

where $F$ represents the path operator *Finally*, indicating that at some state of the path the property will hold, and $p_1$ being the probability that B should follow A within 5 time units. The evaluation of such a path formula in itself is also a state formula which is true at a certain moment in time for the trace. Having defined such state and path formulas, one knows which information to extract from the event log to employ as system states. These system states, along with their case notions and timestamps, then serve as input for the algorithm.

AITIA-PM uses only a subset of PTL by, for example, neglecting the notion of time windows. We do so because long-term dependencies in business processes need to be acknowledged. The interested reader is referred to [13] for more details about PTL.

### 3.2   Algorithmic Procedure

AITIA-PM guides the user in detecting meaningful root causes supported by causal theory. It consists of the following five steps: (i) input data preparation, (ii) generating causal hypotheses, (iii) testing for prima facie causes, (iv) calculation of epsilon values, and (v) testing for causal significance.

**Step 1 – Input Data Preparation.** The AITIA-PM algorithm focuses on *system states* and how they change over *time* for each *case* in the event log. As such, these are the three required attributes in the input data structure. The definition of the system states depends on the potential causes and effects the business user is interested in, and thus, has defined in PTL hypotheses. For example, let's assume that we know that when resource $x$ $(R_x)$ is involved in a case, the case will result in an error $(E)$. In other words, you define your hypothesis as

$$R_x \rightsquigarrow E. \tag{2}$$

Remember that the probability of this leads-to formula actually occurring is inferred from data in a later stage. Given this hypothesis, the data analyst knows which system states to extract from or enrich the event log with: the resources involved with the case at each time unit, and whether or not the error $E$ was registered. As such, the input data consists of these three columns: the case ID, the system state, and the timestamp.

One can also opt to convert all timestamps in the data set to a specific time unit, where the first observation in the event log would start at time unit 0. This would easily allow the reintroduction of time windows in PTL leads-to formulas.

**Step 2 – Generating Hypotheses.** Having defined the system states, one can now generate the different hypotheses: which causes might have a significant impact on the likelihood of the effect triggering? AITIA-PM takes a list of plausible causes and effects to combine them into the complete set of hypotheses: does cause $c$ trigger effect $e$ within the time bounds $[r, s]$? All combinations are considered a hypothesis except where $c = e$.

In this step, it is important to consider adding all system states as a possible cause for the effect of interest. This way, you also check for the other states as potential confounding factors, even though you might not expect them to have a causal relationship with the effect. In the example of $R_x$ triggering an error $E$, a hypothesis will be generated for every resource $R_r$ with $r \in R$ to trigger the effect $E$.

**Step 3 – Testing for Prima Facie Causes.** The hypotheses generated before contain all combinations of cause-effect we are interested in. However, they probably also describe causal relations which might not meet the prima facie condition. In order for a cause to be a prima facie cause of an effect, it must satisfy the following three conditions:

1. the cause must have occurred before the effect,
2. the cause must increase the probability of the effect occurring, and
3. the cause and effect when checking the above requirements must belong to the same case in the event log.

With the timestamps and case IDs provided along with the system states, it is relatively straightforward to determine whether or not a cause is a prima facie cause for an effect from the event log. Only the hypotheses fulfilling the above requirements are considered to be genuine potential causes for the effect.

In order to accomplish this prima facie test, the following pieces of information are required: (i) when and for which case was the cause observed, (ii) when and for which case was the effect observed, and (iii) how often did the effect occur after the cause given they both belong to the same case. The prima facie condition is then probabilistically checked from the data as follows:

$$P(e|c) > P(e) \tag{3}$$

where

$$P(e) = \frac{\#e}{\#events} \tag{4}$$

and

$$P(e|c) = \frac{\#(e \wedge c)}{\#c}. \tag{5}$$

It is important to remember that $\#(e \wedge c)$ takes the timing of events and case ID into account. This computation therefore checks if there exists a $c$ before $e$ within the same case, and if not, the hypothesis is automatically classified as false. For example, resource $R_y$ is only involved after the case already produced error $E$. As such, $P(E|R_y) = 0$, meaning that $R_y$ cannot be a prima facie cause of $E$.

**Step 4 – Calculation of Epsilon Values.** Having determined all prima facie causes of the effect of interest, we now want to separate the genuine causes from the spurious ones. To that end, we use epsilon values as a measure of causality that can be statistically tested. The measure $\epsilon_{avg}$, introduced by Kleinberg [13], describes the average change of probability of effect $e$ given the presence of cause $c$ while keeping another factor $x$ constant. This factor $x$ is also a prima facie cause of $e$ which is deemed to be present. As such, for each other factor $x$, an $\epsilon_x$ is calculated after which the average describes the impact of $c$ on $e$.

Formally, the measure is then expressed as follows:

$$\epsilon_{avg}(c, e) = \frac{\sum_{x \in X \setminus c} \epsilon_x(c, e)}{|X \setminus c|} \tag{6}$$

where $X$ represents the set of prima facie factors of $e$ and

$$\epsilon_x(c, e) = P(e|c \wedge x) - P(e|\neg c \wedge x). \tag{7}$$

Determining these probabilities correctly requires that the case notion is identical for pairs of $e$, $c$ and $x$. While keeping $x$ constant, the probability change of $e$ is of interest when the cause $c$ is present or not. Property (iii) of causality in AITIA-PM dictates that all information regarding causal relationships within a case is available in that same case. As such, the case ID must be identical for $c$ and $x$ when counting the occurrences of $(c \wedge x)$ and $(\neg c \wedge x)$.

The probabilities are defined as follows:

$$P(e|c \wedge x) = \frac{\#(e \wedge c \wedge x)}{\#(c \wedge x)} \tag{8}$$

and

$$P(e|\neg c \wedge x) = \frac{\#(e \wedge \neg c \wedge x)}{\#(\neg c \wedge x)} \tag{9}$$

where $e$ must occur at a later time than $(c \wedge x)$ or $(\neg c \wedge x)$. As soon as this information is available, it is a simple matter of counting how often an effect does or does not take place in the related time windows. For each hypothesis that passed the prima facie test, an $\epsilon_{avg}$ is obtained. These average epsilons are the foundation of the statistical test performed next.

**Step 5 – Determining Causal Significance.** Up until this point, the epsilon values are computed, which express the average probability changes of the effect $e$ occurring given the presence or absence of a prima facie cause $c$. A statistical test can then separate the genuine causes from the spurious ones. To that end, the AITIA-PM algorithm uses the concept of false discovery rates (FDR) as implemented by the R-package `fdrtool` [20]. Saving the technical details, the procedure is as follows:

1. start by calculating z-values: $z = (\epsilon_{avg} - \mu)/\sigma$ where $\mu$ and $\sigma$ represent the average and the standard deviation of the set of $\epsilon_{avg}$, respectively;

2. Next, fit a mixture model to the observed data, the z-values;
3. Determine the FDR of z.

The causal relations where the FDR is below a certain threshold are deemed significant causes. This threshold is chosen freely by the business user depending on how acceptable a false discovery is. For example, with a threshold of 0.01, one would expect 1% of causes to be significant.

## 4  Demonstration

In this section, we demonstrate how AITIA-PM learns causes for process delay by applying it on a real-life dataset, namely the "receipt phase of an environmental permit application process (WABO) CoSeLoG project" event log [4][1]. This event log contains the receiving phase execution records of the building permit application process in an undisclosed Dutch municipality. It consists of 1.434 traces and 8.577 events spread over 27 activity classes.

Similar to Qafari and van der Aalst [17], we consider as effect the delay observed in some cases. This delay threshold is set to 3% of the maximum duration of all traces. As the maximum duration is 275.8813 days, the threshold is equal to 8.2764 days, or 198.6345 h. As the average duration of a trace is about 2% of the maximum duration, the threshold of 3% seems appropriate. We add a new event "Case Delayed" to each case that exceeds the threshold duration at the moment the case reaches a duration of 198.6345 h. This ensures that events occurring after that moment in time can no longer be considered a cause for the delay in that case. As Qafari and van der Aalst [17], we investigate if the combination of a specific activity $A_i$ performed by a specific resource $R_j$ causes process delay.

Remember the five steps of AITIA-PM: (1) data preparation, (2) generating causal hypotheses, (3) testing for prima facie causes, (4) calculation of epsilon values, and (5) testing for causal significance. Steps 1 and 2 both relate to the PTL hypothesis definition. In our example, an initial set of 397 hypotheses is constructed as there are 397 distinct activity-resource pairs in the event log. Each hypothesis for a specific activity $A_i$ and a specific resource $R_j$ can be described with PTL as follows:

$$A_i \land R_j \rightsquigarrow delay \tag{10}$$

Consequently, the system states to extract from the event log are all the activities per case with the associated resource that executed them. The first ten rows of the input dataset are shown in Table 1, along with the first observation of process delay.

All initial 397 hypotheses were tested for the prima facie condition (step 3), and 159 of these passed the test, meaning they occurred before the delay

---

**Table 1.** Input data for AITIA-PM.

| Case ID | System State | Time Unit |
|---|---|---|
| case-891 | Confirmation of receipt - Resource26 | 0.0000000 |
| case-891 | T02 Check confirmation of receipt - Resource26 | 0.0131450 |
| case-891 | T03 Adjust confirmation of receipt - Resource26 | 0.1759917 |
| case-891 | T02 Check confirmation of receipt - Resource26 | 0.1835817 |
| case-891 | T03 Adjust confirmation of receipt - Resource26 | 0.1894819 |
| case-3756 | Confirmation of receipt - Resource02 | 71.2025831 |
| case-3756 | T06 Determine necessity of stop advice - Resource02 | 71.3695931 |
| case-3756 | T02 Check confirmation of receipt - Resource24 | 72.1805186 |
| case-3756 | T07-1 Draft intern advice aspect 1 - Resource24 | 72.1995269 |
| case-3756 | T06 Determine necessity of stop advice - Resource02 | 72.3097125 |
| ... | ... | ... |
| case-891 | Case Delayed | 198.6345127 |
| ... | ... | ... |

was observed and they increase the probability of the case being delayed. After computation of the test statistics and setting the $FDR$ threshold to 5%, we obtain output as shown in Table 2.

**Table 2.** AITIA-PM output.

| cause | epsilon | z | fdr |
|---|---|---|---|
| T02 Check confirmation of receipt - Resource24 | 0.1871651 | 7.444724 | 0.0000000 |
| T04 Determine confirmation of receipt - Resource10 | 0.1202274 | 4.364844 | 0.0258531 |
| T05 Print and send confirmation of receipt - admin1 | 0.0736255 | 2.220631 | 0.0258531 |

In summary, AITIA-PM detects that, with the $FDR$ threshold set to 0.05, three of the 159 hypotheses are genuine. It appears that the probability of the case being delayed significantly increases when specifically (i) "T02 Check confirmation of receipt" is executed by Resource24, (ii) "T04 Determine confirmation of receipt" is executed by Resource10, or (iii) "T05 Print and send confirmation of receipt -" is executed by Admin1. We can be most sure of (i), as that FDR value is equal to zero and its epsilon value is also the highest.

This epsilon is also easy to interpret. In the case of our first result, this interpretation is as follows: the average increase in probability of the effect, the case delay, occurring when the activity "T02 Check confirmation of receipt" is executed by Resource24 while controlling for alternative causal explanations equals 18.71651 pp..

## 5    Conclusion

This paper introduced a novel root cause analysis method in process mining named AITIA-PM. It complements the state-of-the-art with respect to RCA techniques as it follows causality theory. Unlike already established techniques, AITIA-PM imposes realistic assumptions regarding the required data. This makes it a very adaptable technique to the desires of a business user. Additionally, by taking a probabilistic approach and averaging out the probability changes, the technique can easily tackle confounding factors which could cause spurious associations. This makes it a strong novel option for RCA.

The demonstration shows that AITIA-PM can flexibly tap into the vast amount of information an event log possesses. PTL allows very diverse hypotheses to be tested which makes AITIA-PM both powerful but also expressive. Due to PTL it is easy to define both simple as well as more complex hypotheses with respect to cause-effect relations in a formal manner. Finally, we have shown the strength of AITIA-PM with respect to interpretability of results.

Several future research challenges are identified in this article. First, a domain expert is required to provide the necessary states the process can semantically be in. Automatic hypothesis generation could bring insights the domain expert might not even consider. Second, state formulas in their current form are binary as they evaluate to true or false. Future work could bring an extension which supports continuous variables.

## References

1. van der Aalst, W.M.P.: Process Mining: Data Science in Action. Springer, Heidelberg (2016). https://doi.org/10.1007/978-3-662-49851-4
2. van der Aalst, W.M.P., Weijters, T., Maruster, L.: Workflow mining: discovering process models from event logs. IEEE Trans. Knowl. Data Eng. **16**(9), 1128–1142 (2004)
3. Bozorgi, Z.D., Teinemaa, I., Dumas, M., La Rosa, M., Polyvyanyy, A.: Process mining meets causal machine learning: discovering causal rules from event logs. In: 2020 2nd International Conference on Process Mining (ICPM), pp. 129–136 (2020)
4. Buijs, J.: Receipt phase of an environmental permit application process ('WABO'), CoSeLoG project. Eindhoven University of Technology (2014). https://data.4tu.nl/articles/dataset/Receipt_phase_of_an_environmental_permit_application_process_WABO_CoSeLoG_project/12709127

5. Böhmer, K., Rinderle-Ma, S.: Mining association rules for anomaly detection in dynamic process runtime behavior and explaining the root cause to users. Inf. Syst. **90** (2020). Advances in Information Systems Engineering Best Papers of CAiSE 2018

6. Carmona, J., van Dongen, B., Solti, A., Weidlich, M.: Conformance Checking. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-99414-7

7. Delias, P., Lagopoulos, A., Tsoumakas, G., Grigori, D.: Using multi-target feature evaluation to discover factors that affect business process behavior. Comput. Ind. **99**, 253–261 (2018)

8. Ferreira, D.R., Vasilyev, E.: Using logical decision trees to discover the cause of process delays from event logs. Comput. Ind. **70**, 194–207 (2015)

9. Granger, C.W.: Some recent development in a concept of causality. J. Econom. **39**(1–2), 199–211 (1988)

10. Gupta, N., Anand, K., Sureka, A.: Pariket: mining business process logs for root cause analysis of anomalous incidents. In: Chu, W., Kikuchi, S., Bhalla, S. (eds.) DNIS 2015. LNCS, vol. 8999, pp. 244–263. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-16313-0_19

11. Hompes, B.F.A., Maaradji, A., La Rosa, M., Dumas, M., Buijs, J.C.A.M., van der Aalst, W.M.P.: Discovering causal factors explaining business process performance variation. In: Dubois, E., Pohl, K. (eds.) CAiSE 2017. LNCS, vol. 10253, pp. 177–192. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-59536-8_12

12. Hume, D.: A Treatise of Human Nature (1739)

13. Kleinberg, S.: Causality, Probability, and Time. Cambridge University Press, Cambridge (2012)

14. Kleinberg, S., Kolm, P.N., Mishra, B.: Investigating causal relationships in stock returns with temporal logic based methods (2010)

15. Maziarz, M.: A review of the granger-causality fallacy. J. Philos. Econ. Reflections Econ. Soc. Issues **8**(2), 86–105 (2015)

16. Narendra, T., Agarwal, P., Gupta, M., Dechu, S.: Counterfactual reasoning for process optimization using structural causal models. In: Hildebrandt, T., van Dongen, B.F., Röglinger, M., Mendling, J. (eds.) BPM 2019. LNBIP, vol. 360, pp. 91–106. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-26643-1_6

17. Qafari, M.S., van der Aalst, W.: Root cause analysis in process mining using structural equation models. In: Del Río Ortega, A., Leopold, H., Santoro, F.M. (eds.) BPM 2020. LNBIP, vol. 397, pp. 155–167. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-66498-5_12

18. Qafari, M.S., van der Aalst, W.M.P.: Case level counterfactual reasoning in process mining. In: Nurcan, S., Korthaus, A. (eds.) CAiSE 2021. LNBIP, vol. 424, pp. 55–63. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-79108-7_7

19. Fani Sani, M., van der Aalst, W., Bolt, A., García-Algarra, J.: Subgroup discovery in process mining. In: Abramowicz, W. (ed.) BIS 2017. LNBIP, vol. 288, pp. 237–252. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-59336-4_17

20. Strimmer, K.: fdrtool: a versatile R package for estimating local and tail area-based false discovery rates. Bioinformatics **24**(12), 1461–1462 (2008)

21. Suppes, P.: A probabilistic theory of causality. Br. J. Philos. Sci. **24**(4), 409–410 (1973)

22. Suriadi, S., Ouyang, C., van der Aalst, W.M.P., ter Hofstede, A.H.M.: Root cause analysis with enriched process logs. In: La Rosa, M., Soffer, P. (eds.) BPM 2012. LNBIP, vol. 132, pp. 174–186. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-36285-9_18

23. Vasilyev, E., Ferreira, D.R., Iijima, J.: Using inductive reasoning to find the cause of process delays. In: 2013 IEEE 15th Conference on Business Informatics, pp. 242–249. IEEE (2013)
24. Vogt, W.P., Johnson, B.: Dictionary of Statistics & Methodology: A Nontechnical Guide for the Social Sciences. Sage, Thousand Oaks (2011)