CC-HIT: Creating Counterfactuals from High-Impact Transitions

Xian Zhi-Cong[1,2], Ludwig Zellner[1,2], Gabriel Marques Tavares[1,2], and Thomas Seidl[1,2]

[1] Database Systems and Data Mining, LMU Munich, Munich, Germany
[2] Munich Center for Machine Learning, Munich, Germany
{xian, zellner, tavares, seidl}@dbs.ifi.lmu.de

Abstract. Reliable process information, especially regarding trace durations, is crucial for smooth execution. Without it, maintaining a process becomes costly. While many predictive systems aim to identify inefficiencies, they often focus on individual process instances, missing the global perspective. It is essential not only to detect where delays occur but also to pinpoint specific activity transitions causing them. To address this, we propose CC-HIT (Creating Counterfactuals from High-Impact Transitions), which identifies temporal dependencies across the entire process. By focusing on activity transitions, we provide deeper insights into relational impacts, enabling faster resolution of inefficiencies. CC-HIT highlights the most influential transitions on process performance, offering actionable insights for optimization. We validate this method using the BPIC 2020 dataset, demonstrating its effectiveness compared to existing approaches.

Keywords: Local Explanation · Impact Ranking · Shapley Value · Counterfactuals.

1 Introduction

Improving processes to exploit their full potential is critical to survive in a competitive market. When processes are optimized, they may use fewer resources, improve their output, and decrease their duration, leading to better overall performance [3]. This way, identifying performance issues is important for smooth executions. Current solutions focus their explanations and recommendations on a trace-level scope, which causes them to miss the overall view of processes [2, 16, 18].

When we take temporal optimization of processes as an example we might ask ourselves the following questions: *Which process paths entail the highest delays and lags? Which activities are involved? Are there specific activity transitions that are to blame?* The order of these questions reveals an important aspect: although it is essential to identify objectively slow traces, pinpointing interdependent activities, i.e., specific activity transitions, is much more beneficial. A reason for that, on the one hand, is the multiple occurrences of these activity combinations in different process paths influencing multiple traces. On the

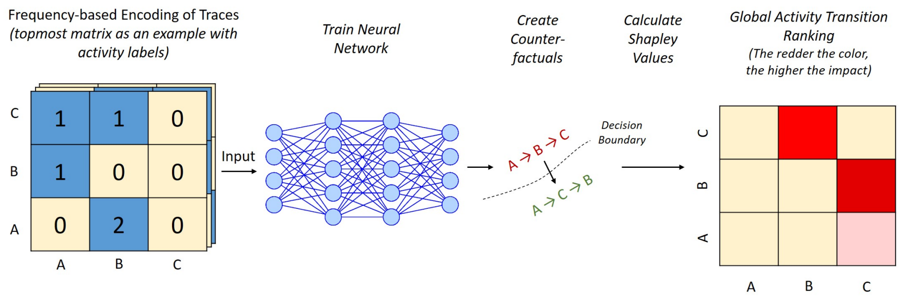

Fig. 1. Main idea of CC-HIT: After training the neural network on the transition matrices of each trace, counterfactuals are generated. Using these counterfactuals, Shapley values are computed for each transition between activity pairs, leading to a global ranking of activity transitions.

other hand, a transition can manifest itself differently in various traces leading to different impacts. From the second reason follows that pinpointing and explaining transitions locally is imperative to understand the source of inefficiency. However, the key point involves maintaining a global view, as these insights are drawn from it.

Key performance indicators (KPIs), which can be based on factors like time or cost, are used to measure the success of specific entities in relation to these attributes. Using CC-HIT, we analyze processes based on a chosen KPI, giving full control of the analysis to the process owner. With this, we manage to derive the impact of activity transitions for a trace regarding the specified KPI, i.e., we handle the event log as input and produce a ranking of transitions, showing how each one impacts the trace. Additionally, CC-HIT provides alternative insights in the event log by incorporating counterfactuals [13, p. 262]. Hence, we explore how changing certain factors would affect outcomes and use these insights to provide a global activity transition ranking. Figure 1 demonstrates the steps of our approach. We start by training a model using the activity transition data from all traces in the event log. This helps the model learn patterns and relationships between different activities. Next, we create hypothetical scenarios, i.e. counterfactuals, to explore how changing certain factors might affect the results. We then use these counterfactuals to calculate Shapley values, which measure the impact of each activity transition on KPIs. This procedure precisely identifies activity transitions that are high-impacting on KPIs providing deeper insights than typical shallow analysis.

In this paper, we first review related work to contextualize our research and identify existing gaps (Section 2). We then present our preliminary section (Section 3), which covers the theoretical foundations and definitions pertinent to our study. Our methodology section details the research design and analysis techniques employed (Section 4). Following this, we describe our experiments, present and analyze the results, and evaluate our findings (Section 5). Finally, we conclude with a summary of our key findings and suggest directions for future research (Section 6).

2 Related Work

Explainable AI (XAI) techniques have been extensively applied in event data. Special focus is observed in predictive and prescriptive tasks aiming to forecast future outcomes (e.g., remaining time, next-activity) [4] and to recommend actions [14]. For instance, Pauwels and Calders [16] take advantage of Bayesian Networks to provide reasoning behind predictions. Galanti et al. [7] combined a machine learning KPI predictor with Shapley values to leverage explanations. Similarly, a post-hoc explanation is obtained by the application of Local Interpretable Model-Agnostic Explanations (LIME) in [18]. As performance in predictive tasks is improved with deep learning models, Mehdiyev and Fettke [12] tailored a method for local post-hoc analysis for deep neural networks. Concerned with cross-assessment, El-khawaga et al. [6] introduced a framework for comparison of local and global XAI methods for predictive tasks with a focus on empirical evaluation. Additionally, [19] proposed a blockchain-based auditing system that utilizes maximum-likelihood evidential reasoning to attribute the impact of legal facts in legal documents to a final decision. However, its applicability to process mining must be investigated, as legal documents and event data have different intrinsic characteristics, such as logical reasoning based on closed-world assumptions.

In the realm of prescriptive process monitoring, Bozorgi et al. [2] applied Shapley values on top of a causal effect approach. For that, the authors fix the prefix vector representation to assess feature importances and provide an analysis of causalities. Stevens et al. [21] investigated the trade-off between interpretable models versus model-agnostic XAI techniques and introduced a notion of explainability that allows comparisons of methods from different families.

Although XAI has been extensively used in event data, its application is limited to predictive tasks for providing explanations that support model decisions, i.e., to justify model choices. Moreover, these approaches work on trace prefixes since the task is to predict forthcoming steps [8], [15]. Contrarily, our approach aims at looking at historical event data to uncover hidden patterns that influence on KPI performance. Thus, supporting more informed decision-making within organizations.

3 Preliminaries

This section covers the definitions regarding process mining and Shapley value applications.

Definition 1 (Activity, Event). *An event contains the information about an activity occurring at a specific time in a specific context. Additionally, it can* *provide further data attributes. Here, we refer to the realm of activities by $\mathcal{A}$. An event is defined as a tuple* $\mathrm{e} = (\mathrm{c}, \mathrm{a}, \mathrm{t}) \in (\mathbb{N}, \mathcal{A}, \mathbb{N})$ *consisting of a case identifier* c*, an activity* a*, a timestamp* t*. The universe of all possible events is denoted by* $\mathcal{U}_e$*.*

An ordered, finite sequence of events constitutes a case, whereas an event log contains multiple cases.

Definition 2 (KPI). *We assign a key performance indicator (KPI) $\mathcal{K}$ to each case. This numerical value can represent any case attribute, such as time, money, or other measurable attributes.*

Definition 3 (Activity Transition). *Given our set of activities $\mathcal{A}$ and a transition relation $\mathcal{T}$ being a subset of $\mathcal{A} \times \mathcal{A}$. We define a transition from a_1 to a_2 iff $(a_1, a_2) \in \mathcal{T}$.*

In our work, we calculate Shapley values [20] by treating each activity transition as a player.

Definition 4 (Contribution, Coalition). *A subset of players is called a coalition S where the set of possible coalitions corresponds to the powerset of players $\mathcal{P}(\mathcal{N})$. A contribution function v maps a subset of players to the real numbers $v : 2^{\mathcal{N}} \to \mathbb{R}$, with $v(\emptyset) = 0$, where $\emptyset$ denotes the empty set. Hence, given v the calculation of $v(S)$ yields the contribution of the coalition S.*

We utilize a transition matrix where each cell represents the transition value of each pair in $(a_i, a_j) \in \mathcal{M}$. This matrix helps us capture and analyze the transition value between consecutive events regarding its respective attribute $\mathcal{K}$. For instance, if we assign time as $\mathcal{K}$, we derive the value for an event transition as the time difference between two consecutive events $\Delta_t(e^i, e^{i+1}) = t_{e^{i+1}} - t_{e^i}$.

Definition 5 (Shapley Value [20]). *The Shapley value is defined by the following equation:*

$$\phi_j(v) = \sum_{S \subseteq \mathcal{A} \setminus \{a_j\}} \frac{|S|!(|\mathcal{A}| - |S| - 1)!}{|\mathcal{A}|!} (v(S \cup \{a_j\}) - v(S))$$

Hence, $\forall a_j \in \{a_k\}_{k=1}^{|\mathcal{A}|}$ we calculate the average marginal contribution to every possible coalition which eventually yields the impact for each activity transition regarding $\mathcal{K}$.

The exact relationship between activity transitions and the final case KPI in cases is often unknown and usually surrogate models are used to approximate it. One such possibility is to use machine learning prediction models.

Definition 6 (Prediction models for activity transitions). *The oracle function $\Phi : \mathcal{E} \to \mathcal{K}$ maps a case containing activity transitions to its KPI. For a case, σ with activity sequences, we can derive a list of activity transitions $\sigma' = \langle (a_1, a_2), (a_2, a_3), \cdots, (a_{|\sigma'|-1}, a_{|\sigma'|}) \rangle$. A prediction model $f : \mathcal{E}' \to \mathcal{K}$ maps activity transitions in a case to its KPI. It is an approximation of the oracle function Φ, i.e., $f \approx \Phi$.*

4 Methodology

We assume the activities happen instantaneously (atomic event assumption) and apply the Markov assumption, which states that the probability of the next activity occurring, given the current and past activities, depends only on the current activity [10]. In the scope of this paper, we mainly focus on the types of activities and their inter-relationships that can contribute to the case KPI $\mathcal{K}$.

4.1 Activity transition matrix

In process mining, representing the sequential orders of activities is crucial for understanding how processes flow and identifying inefficiencies or deviations. We propose using transition matrices to represent the interaction among activities. Given a total number of activities $|\mathcal{A}|$ as N, we construct a matrix M of size $N \times N$ with preceding activities represented in the rows and succeeding activities represented in the columns. Furthermore, each entry $M_{i,j}$ is defined as follows:

$$M_{i,j} = \begin{cases} n, & \text{if there are transitions from activity } i \text{ to activity } j \text{ in one case} \\ & \text{then the number of these transitions} \\ 0, & \text{otherwise} \end{cases} \tag{1}$$

Multiple occurrences of the same activity within a single case are possible in an event log. As a result, there may be different subsequent activities for a single activity, leading to several entries in the same column of the transition matrix. Additionally, by representing the temporal sequencing of activities as a transition matrix, we can easily observe self-loops on the diagonal matrix, indicating instances where activities transition to themselves. Another benefit of this representation is that it encodes the temporal relationships among activities. It allows us to identify infeasible transitions as zero entries in the transition matrix across the entire event log. This matrix representation resembles an image, facilitating the use of machine learning models for predicting case KPIs.

Consequently, we treat the transition matrices as two-dimensional images and apply deep neural networks to predict the final case KPI. We then derive estimations for coalition functions from the trained model to compute Shapley values.

4.2 The Shapley additive explanation framework (SHAP)

As described in Def. 5, the computation of Shapley values requires estimating coalition values $v(S)$. When using machine learning models to approximate the relationship between case activity transitions and their KPIs, it is important to address the issue of missing activity transitions that only appear in some cases within the event log. Lundberg et al. [11] addresses this challenge by proposing the SHapley Additive exPlanation framework (SHAP).

In this framework, the coalition values for player subsets are computed as the conditional expectation of the outcome given these player subsets marginalizing over all the other missing players, i.e., $v(S) = \mathbb{E}[f(x)|x_S]$. x are the input features to our prediction model f, equal to the activity transitions $(a_i, a_j) \in \mathbf{M}$ in event logs with x_S denoting the activity transitions defined in the player subset S. Since trace activity transitions resemble 2D images, and we aim to extract both local features, such as adjacent activity transitions, and global features, such as distant influences, we employ a residual neural network (ResNet) [9] to obtain the prediction function $f(x)$.

A common practice in SHAP for estimating coalition values for player subsets is to predict the outcomes for these subsets while replacing missing players with mean values from background datasets. The resulting Shapley values then explain the input predictions relative to these background datasets [1]. The background datasets are selected by uniformly sampling from the given datasets. However, in process mining, our focus extends beyond identifying which activity transitions increase overall case throughput time on average; we are also interested in actionable insights for improving process KPIs. Assume a dataset with two features, x_1 and x_2. To estimate x_1's contribution to the predicted outcome, we compare samples with actual x_1 values to those with dummy x_1 values, often set as the average x_1. However, this may be unrealistic since x_1 could depend on x_2, and different x_2 values may restrict the range of x_1. Albini et al. [1] proposed using counterfactual Shapley values to improve the predicted outcomes. We will adapt this approach for process mining in the next section.

4.3 The generation of counterfactuals

While standard Shapley value computations assign importance scores to activity transitions based on an average case—such as one with average frequencies of all occurring activity transitions—in the given event logs, this average case may not be viable in the real world. A case may not necessarily experience all feasible activity transitions, as some are mutually exclusive. Standard Shapley values do not capture this limitation. To address this, we propose using counterfactuals—minimal feasible changes in the sequence of activities that can lead to significant deviations in case KPIs—as reference cases. This section focuses on generating counterfactuals for process mining. Consider a loan approval process with the following sequence of activities: "Application Submitted High/Medium/Low Loan" (A) → "Automated/Manual Document Verification" (B) → "Initial Automated/Manual Credit Check Performed" (C) → "Loan Officer Detailed Risk Assessment" (D) → "Loan Offer Generated" (E) → "Loan Decision (Approved/Declined)" (F) → "Customer Notification" (G), and we aim to generate counterfactuals to accelerate the loan application process. Before introducing our algorithm to generate counterfactuals, we must first examine the properties that qualify as good counterfactuals in process mining:

- *Plausibility.* Some activity transitions are mandatory, and certain steps must occur in a specific order. In the previous loan application example, having C occuring B before does not make sense because credit checks should only be performed on verified documents.

- *Proximity.* The generated counterfactual must be similar to the sample of interest. In the loan approval process, a minimal change in the activity sequence, such as swapping C and D, can lead to a significant deviation in the runtime. If the D occurs before C, manual effort is wasted as the officer reviews the application without the necessary credit information. Reordering these two activities can substantially impact the process runtime, making this change a valuable counterfactual.
- *Feasibility.* Some transition changes are only possible depending on previous workflows. In the previous loan approval application, a "Pre-Approved Loan Offer" can be generated if B is completed, no discrepancies are found, and activity C identifies the application as low-risk. If any documents are missing or incorrect or the loan amount is large, the process moves to a manual review, and pre-approval is not possible.

To satisfy the aforementioned properties, we propose generating counterfactuals by projecting onto decision boundaries based on constraints and along directions derived from the principal components of the nearby neighborhood. We describe this method in the following steps and apply it to the loan application process introduced at the beginning of this section to enhance understanding:
Step 1: Definition of the counterfactual outcome: If the prediction outcome is continuous, we categorize it into discrete labels such as desired and undesired. For instance, if the loan application considers throughput time as its KPI $\mathcal{K}$, it could be categorized as fast-tracked or delayed. If the queried samples are delayed, we define counterfactual outcomes as fast-tracked or on time.
Step 2: Obtaining immutable constraints: Creates a binary mask matrix for activity transitions, with zeros for infeasible transitions and ones for feasible ones. Since the steps of the loan application process are causal, we can mask reverse orderings of steps with zeros for infeasible changes.
Step 3: Initial generation of counterfactual samples: Randomly sample data and use prediction models to select candidates that match the counterfactual outcomes. We randomly draw samples and resample for the loan application process until we have enough instances with no delays.
Step 4: Finding K-nearest neighbors: To determine feasible directions for altering samples to their counterfactuals, we first identify the nearest neighbors for the queried samples. For example, clustering delayed samples may reveal that they all have "Application Submitted HIGH Loan". In contrast, other samples that follow similar steps, such as "Automated/Manual Document Verification" $\rightarrow$ "Initial Credit Check Performed", have a lower overall run-time due to a smaller loan amount.
Step 5: Calculation of principal axes of the clusters: We derive feasible changes in the activity transitions by analyzing their correlations. Mathematically, this involves performing Principal Component Analysis (PCA) on the clustered data near the query points to capture correlations among activity transitions. For instance, the previous step shows a strong correlation between the requested loan amount and final throughput time. Thus, one feasible change is to decrease the requested loan amount.

Step 6: Projection of the counterfactual candidates onto decision boundary along the principal axes of the clusters: Iteratively project the difference between the query data points and counterfactual candidates onto the principal components, gradually pulling the counterfactual candidates closer to the query sample points until they are near the decision boundary. In the loan application process, this means successively reducing the loan amount to align with cases with a faster approval track.

5 Evaluation

In this section, we focus on the impact of activity transitions on the throughput time as the KPI. For our experiments, we use the permit log from the BPIC Challenge 2020 event log [5], focusing on throughput time as the key KPI. We aim to identify potential activity transitions for improvement.

5.1 Dataset description

The travel permit log documents the billing process at the Eindhoven University of Technology from 2017 to 2018. The event log contains 7,065 cases, some of which are also included in the international declaration event log. This overlap occurs because international trips require supervisor permission, obtained by filing a travel permit. This permit must be approved before any travel arrangements are made. Statistical analysis reveals an average throughput time of 87.4 days compared to a median of 71.73 days. The maximum and minimum throughput times are 1190 and 0.53 days, respectively. This indicates that the case duration is slightly skewed across the entire event log, with some cases experiencing significant delays.

A standard process workflow typically proceeds as follows: employees submit travel requests, which are then forwarded to budget owners, supervisors, directors, and finally, administrative departments for approval. Once travel is granted, employees can begin their trips. After the trip ends, employees must submit travel declarations detailing their expenses, followed by a request for payment to cover these expenses. The approval procedures for travel declarations and payment requests follow similar steps to those for travel approval. Pufahl et al. [17] find out that the event log conforms to the standard procedure in most cases.

5.2 Experiment results

Most methods discussed in Section 2 focus on trace-level assessments using case or activity attributes while we uniquely study the impact of activity transitions. To evaluate and compare different methods for analyzing activity transitions, we visualize various plots that rank the importance of these transitions. Additionally, we discuss the additional insights our approach provides by directly incorporating throughput time into the analysis.

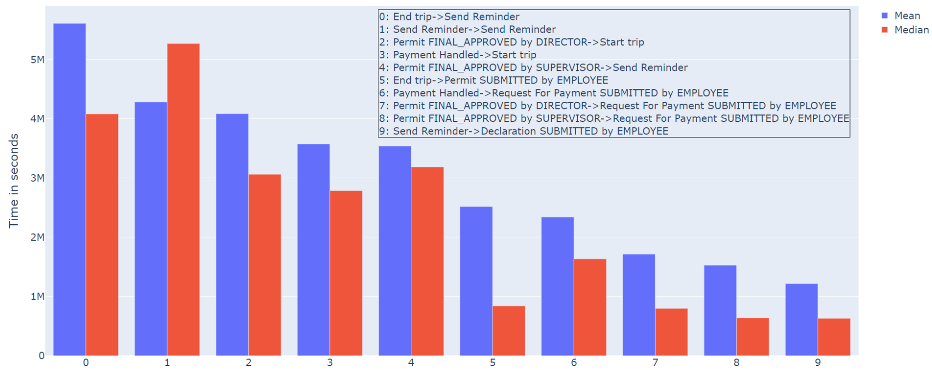

Fig. 2. Ranking of the activity transitions based on mean and median duration.

Baseline. The classical approach to analyzing the importance of activity transitions on the final throughput time involves calculating the mean and median for each possible activity transition across the event log. The event log contains 51 unique activities, resulting in 51×51 possible activity transitions. We then filter out the infrequent transitions to reduce the number of transitions to consider. Next, we rank the remaining activity transitions according to their *median* duration and select the top 10 for visualization. As shown in Fig. 2, the activity transition "Send Reminder" $\rightarrow$ "Send Reminder" consumes the most time, with the transition "End trip" $\rightarrow$ "Send Reminder" in the second place. We observe that transitions involving the "Send Reminder" activity significantly impact the throughput, which makes sense because a reminder will only be sent when the expected next activity is pending. Other activities, such as "Start trip" and "End trip", are also associated with high-duration transitions. However, from Fig. 2, we cannot determine the expected next activity to prevent the system from sending a reminder, even when examining transitions with minimal duration.

CC-HIT. Unlike traditional analysis, which only considers the absolute activity transition time in the event log, our approach encodes the activity transitions to predict the final throughput time and uses the prediction model to assess the impact of these transitions compared to what-if scenarios. We then quantify these impacts using Shapley values. If changing the state of an activity transition from absence to presence (or vice versa) reduces throughput time, that activity transition will have a higher Shapley value. Since the original SHAP framework provides instance-wise interpretation, and we are focused on the overall Shapley values in the event log, we average these values across the entire event log and rank the activity transitions based on the mean of their Shapley values. This allows us to generate a bee swarm plot, as shown in Fig. 3. As depicted in Fig. 3, the activity transitions are shown on the x-axis. At the same time, each point represents the Shapley value for a specific activity transition, plotted along the y-axis. The color of the points in the figure represents the original transition encoding, with zero values indicating absence and positive values indicating presence.

Compared to Fig. 2, the activity transition that now has the most significant impact on the final throughput time is the submission of payment requests to the administration, i.e., “Request For Payment SUBMITTED by EMPLOYEE” → “Request for Payment APPROVED by ADMINISTRATION”. Indeed, when we analyze the entire event log and calculate the case throughput with and without this activity transition, we find an average difference of 22.9 days.

Another interesting finding is that the presence of an activity transition does not automatically result in an average increase in the final throughput time; it can also lead to a decrease. Let us consider the transition from “End trip” to “Declaration SUBMITTED by EMPLOYEE” as an example. Its presence, marked in blue, can reduce the final throughput time, as depicted by the negative Shapley values. This occurs because employees should submit their declaration immediately after the trip. If they do not, the system will send a reminder, which increases the final throughput. This is illustrated by the activity transition from “End trip” to “Send reminder”, which ranks ninth in terms of throughput impact.

Another advantage of our approach is that less frequent activity transitions, which may be time-consuming, do not necessarily have a high impact on changing throughput time. For example, the transition from “Permit FINAL_APPROVED by DIRECTOR” to “Start trip”, while present in Fig. 2, does not appear prominently in our analysis. This is because it occurs infrequently in only 362 out of 7,065 cases. Therefore, our approach provides a more robust and insightful view of the impact of activity transitions, shining light on influential transitions and further supporting decision-makers in improving their processes.

However, when we consider the transition from “End trip” to “Permit SUBMITTED by EMPLOYEE” as an example, we see that submitting a permit after the trip can speed up the process. Nonetheless, this may introduce other risks, such as the potential rejection of the reimbursement. Therefore, a reduction in throughput time does not necessarily indicate an optimal outcome, as our prediction model does not account for other risks, such as final activity labels like rejection. Consequently, it remains up to the process owner to determine which transition links can be improved.

Nevertheless, our data-driven approach provides valuable insights based on the data, reflecting the practices and outcomes observed.

6 Conclusion

Process performance is often measured by KPIs. However, identifying transitions that influence process outcome is challenging. Standard approaches focus on instance-level analysis or assess impact based on mean throughput time. We propose CC-HIT, a framework for identifying influential activity transitions that stakeholders can further improve once identified. We compared CC-HIT to traditional statistical analysis and demonstrated its advantages. Our approach

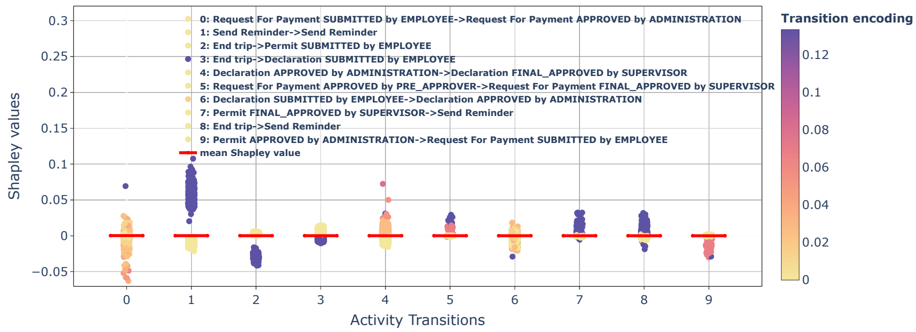

Fig. 3. Ranking of the activity transitions based on Shapley values referencing counterfactual cases.

identifies transitions that differentially impact the final process KPIs and shows how the presence or absence of these transitions can influence the final KPI. Our method is more data-driven and provides deeper insights than traditional statistical approaches. While we aggregate Shapley values by their average to obtain a global assessment of activity transitions, we still retain local information, such as the distribution of Shapley values across instances for each activity transition, as visualized in Fig. 3. A product owner can design additional metrics, such as incorporating the variance of Shapley values, to aggregate individual impacts. Nonetheless, our approach highlights all significant transitions worth noting and offers opportunities for future process optimization. In future work, we aim to use the identified transitions to support stakeholders in making direct decisions, suggesting potential ways to overcome bottlenecks.

References

1. Albini, E., Long, J., Dervovic, D., Magazzeni, D.: Counterfactual shapley additive explanations. In: Proceedings of the 2022 ACM Conference on Fairness, Accountability, and Transparency. p. 1054–1070. FAccT '22, Association for Computing Machinery, New York, NY, USA (2022)
2. Dasht Bozorgi, Z., Teinemaa, I., Dumas, M., La Rosa, M., Polyvyanyy, A.: Prescriptive process monitoring based on causal effect estimation. Information Systems **116**, 102198 (2023)
3. Di Cunzolo, M., Guastalla, A., Aringhieri, R., Sulis, E., Amantea, I.A., Ronzani, M., Di Francescomarino, C., Ghidini, C., Fonio, P., Grosso, M.: Combining process mining and optimization: A scheduling application in healthcare. In: International Conference on Business Process Management. pp. 197–209. Springer (2022)
4. Di Francescomarino, C., Ghidini, C.: Predictive Process Monitoring, pp. 320–346. Springer International Publishing, Cham (2022)
5. van Dongen, B.: Bpi challenge 2020 (2020). `https://doi.org/10.4121/UUID:52FB97D4-4588-43C9-9D04-3604D4613B51`

6. El-khawaga, G., Abu-Elkheir, M., Reichert, M.: Xai in the context of predictive process monitoring: An empirical analysis framework. Algorithms **15**(6) (2022)
7. Galanti, R., Coma-Puig, B., Leoni, M.d., Carmona, J., Navarin, N.: Explainable predictive process monitoring. In: 2020 2nd International Conference on Process Mining (ICPM). pp. 1–8 (2020)
8. Galanti, R., de Leoni, M., Monaro, M., Navarin, N., Marazzi, A., Di Stasi, B., Maldera, S.: An explainable decision support system for predictive process analytics. Eng. Appl. Artif. Intell. **120**(C) (apr 2023)
9. He, K., Zhang, X., Ren, S., Sun, J.: Deep Residual Learning for Image Recognition. In: Proceedings of 2016 IEEE Conference on Computer Vision and Pattern Recognition. pp. 770–778. CVPR '16, IEEE (Jun 2016)
10. Helske, J., Helske, S., Saqr, M., López-Pernas, S., Murphy, K.: A Modern Approach to Transition Analysis and Process Mining with Markov Models in Education, pp. 381–427. Springer Nature Switzerland, Cham (2024)
11. Lundberg, S.M., Lee, S.I.: A unified approach to interpreting model predictions. In: Guyon, I., Luxburg, U.V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., Garnett, R. (eds.) Advances in Neural Information Processing Systems. vol. 30. Curran Associates, Inc. (2017)
12. Mehdiyev, N., Fettke, P.: Explainable Artificial Intelligence for Process Mining: A General Overview and Application of a Novel Local Explanation Approach for Predictive Process Monitoring, pp. 1–28. Springer International Publishing, Cham (2021)
13. Molnar, C.: Interpretable machine learning. Lulu. com (2020)
14. Padella, A., de Leoni, M., Dogan, O., Galanti, R.: Explainable process prescriptive analytics. In: 2022 4th International Conference on Process Mining (ICPM). pp. 16–23 (2022)
15. Pasquadibisceglie, V., Appice, A., Ieva, G., Malerba, D.: TSUNAMI - an explainable PPM approach for customer churn prediction in evolving retail data environments. Journal of Intelligent Information Systems **62**(3), 705–733 (Jun 2024)
16. Pauwels, S., Calders, T.: Bayesian network based predictions of business processes. In: Fahland, D., Ghidini, C., Becker, J., Dumas, M. (eds.) Business Process Management Forum. pp. 159–175. Springer International Publishing, Cham (2020)
17. Pufahl, L., Hobeck, R., Binetruy, P., Chada, W., Mykola, D., Gülle, K.J., Slarzynska, M., Stiehle, F., Weber, I.: Performance variant and conformance analysis of an academic travel reimbursement process. In: BPI Challenge 2020 (2020)
18. Rizzi, W., Di Francescomarino, C., Maggi, F.M.: Explainability in predictive process monitoring: When understanding helps improving. In: Fahland, D., Ghidini, C., Becker, J., Dumas, M. (eds.) Business Process Management Forum. pp. 141–158. Springer International Publishing, Cham (2020)
19. Sachan, S., Liu (Lisa), X.: Blockchain-based auditing of legal decisions supported by explainable ai and generative ai tools. Engineering Applications of Artificial Intelligence **129**, 107666 (2024)
20. Shapley, L.S., et al.: A value for n-person games (1953)
21. Stevens, A., De Smedt, J., Peeperkorn, J.: Quantifying explainability in outcome-oriented predictive process monitoring. In: Munoz-Gama, J., Lu, X. (eds.) Process Mining Workshops. pp. 194–206. Springer International Publishing, Cham (2022)