Hypothesis Testing for Processes

Cameron Pitsch*, Tobias Brockhoff*, Jan Niklas Adams*,
Sander J. J. Leemans*, Leo Anthony Celi†, Wil M. P. van der Aalst*
*RWTH Aachen University, Aachen, Germany
{cameron.pitsch, brockhoff, niklas.adams, wvdaalst}@pads.rwth-aachen.de, s.leemans@bpm.rwth-aachen.de
†Massachusetts Institute of Technology, Cambridge, MA, USA
lceli@mit.edu

***Abstract*—Process mining techniques are useful for analyzing and optimizing processes. However, processes often exist in many variants that can differ significantly in their execution. These differences within the data can negatively affect the quality of process mining results and, furthermore, indicate disparities within the process. While several approaches exist that characterize the differences between processes, oftentimes the mere existence of differences can be problematic. To this end, techniques to prove or disprove the existence of such differences are required and should do so in a statistically sound manner. However, the literature on process hypothesis testing is sparse and limited in the considered dimensions of difference. In this paper, we propose a hypothesis testing approach that uses the earth mover's distance in combination with a permutation test to compare event logs in various dimensions. The evaluation shows that the proposed approach achieves better performance than the existing work in detecting control-flow differences and, moreover, detects differences in further dimensions, demonstrated on the time dimension.**

***Index Terms*—Process Mining, Process Hypothesis Testing, Earth Mover's Distance**

I. Introduction

In process mining [1], event data recorded during the execution of processes is analyzed to extract knowledge about the underlying process. However, the nature of the process is not necessarily the same for all cases. For instance, processes executed in different parts of the world are subject to different laws and, as such, may require additional checks and approvals. These differences between process variants cause a heterogeneity in the event data, which can negatively affect the quality of process mining results. Moreover, these differences can be a symptom of disparities within the process. Techniques exist that compare processes and provide diagnostics describing differences in their execution, for instance, by means of an annotated process model. However, in some contexts, the mere existence of such differences poses an issue. In this regard, the ability to prove or disprove the existence of such differences in a sound manner is vital. For instance, in 2023, the Biden administration of the United States of America introduced a law to promote equitable healthcare by rewarding Medicare providers that are able to show a high level of health equity [2]. *Process Hypothesis Testing* (PHT) techniques take a statistical approach to performing such comparisons, providing a p-value as output which can be used to determine if a statistically significant difference exists between the compared processes. However, only one PHT approach exists [3], which is limited to comparisons in the control flow. Process comparison techniques are focused on providing diagnostics and characterizing differences between processes, and require human interpretation. Furthermore, existing techniques are often limited in the considered dimensions and their ability to provide statistical guarantees [4]–[8]. In this regard, an ideal PHT technique would fulfill the following requirements:

- **R1)** *Control-Flow Aware:* The approach considers the control flow in its comparison.
- **R2)** *Multidimensional:* The approach considers further dimensions in the comparison.
- **R3)** *Statistically Sound:* The approach uses statistical tests to detect differences with statistical guarantees.
- **R4)** *Holistic Comparison:* The approach compares the processes as a whole, as opposed to performing many local comparisons, for instance, on individual directly follows relationships.

In this paper, we present a process hypothesis testing technique that addresses all of these goals. It compares processes in multiple dimensions in a statistically sound manner, providing a p-value as output. In particular, we implement the approach for comparisons regarding the control flow and timed control flow. An overview of the method is shown in Figure 1. As input, the approach takes two event logs, from which representations are extracted which describe the aspects of the process that we compare. Then, we compute pairwise distances between these representations and perform a permutation test using the Earth Mover's Distance (EMD). The output of the approach is a p-value, which can be used to report a difference if it is sufficiently small. In our evaluation, we find that the proposed approach achieves a better Type I error rate than the existing work in [3], and performs well in detecting differences in further dimensions.

The remainder of the paper is structured as follows: In Section II, we discuss the related work on process comparison and process hypothesis testing. Section III introduces the fundamental concepts required to reason about the approach. In Section IV, we discuss the proposed method and its relation to the existing work in [3]. Next, we evaluate the method and compare its performance to existing work in Section V.

The authors gratefully acknowledge the financial support by the Federal Ministry of Education and Research (BMBF) for the joint project AIStudyBuddy (grant no. 16DHBKI016).

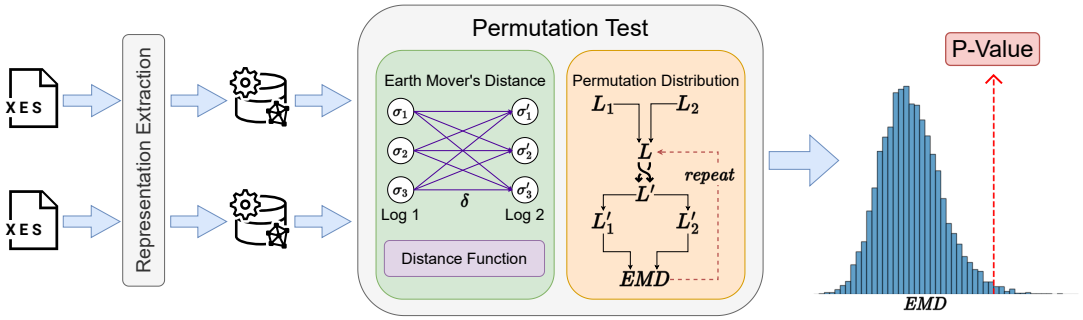

Fig. 1. An overview of the proposed method. After extracting representations of the cases in the event logs, pairwise distances are computed using a distance function, and a permutation test is performed using the earth mover's distance as a test statistic.

Finally, in Section VI, we conclude the paper and discuss limitations and future work.

II. Related Work

Operational processes can be compared using three approaches: *model-to-model*, *log-to-log*, and *log-to-model*. While log-to-model comparisons [9], [10], also known as conformance checking, compare event data against a process model to locate deviating behavior, model-to-model comparisons [11], [12] are concerned with identifying differences between process models. In contrast, the log-to-log perspective, which is the one taken in this paper, considers event data as the source of truth. This means that the comparison is based on the actual execution of the process. Table I gives an overview of the related work in log-to-log comparisons and hypothesis tests, and categorizes them based on the requirements from Section I. In general, the focus of process comparison techniques does not lie in proving the existence of differences, but rather characterizing them and providing diagnostics. As such, by their design, their applicability for our use case is limited. Nonetheless, since only one PHT approach exists, we extend the scope of related work to cover process comparison. While all approaches consider the control flow in their comparison (**R1**), only about half of the approaches go beyond this and include further dimensions (**R2**). Similarly, less than half of the approaches involve statistical tests (**R3**), and only two of the approaches perform a holistic comparison and return a single measure as output (**R4**). The approach of Leemans et al. [3] is the only approach to perform a single, holistic, statistical test returning a p-value output. For this reason, it is currently, to the best of our knowledge, the only PHT approach.

TABLE I
Related work on log-to-log comparisons and hypothesis tests, and which requirements from Section I they fulfill.

Related Work	R1	R2	R3	R4
[4]–[8]	✓			
[13]–[17]	✓	✓		
[18]–[23]	✓	✓	✓	
[24]	✓		✓	
[25]	✓			✓
[3]	✓		✓	✓

Several approaches give an augmented process model as output, giving a visual overview of the differences between the processes, as well as the magnitude of the differences. Ballambettu et al. [14] discover process maps from event logs and annotate them with process metrics. Then, the differences in these metrics between the event logs are computed and visualized using node and arc thickness. Similarly, Bolt et al. [21], [22] discover annotated transition systems and compare the annotation values using a two-tailed *Welch's T-Test*. Then, significant differences are visualized using thickness and color to convey the mean annotation value and the effect size, measured using *Cohen's d*, respectively. Cecconi et al. [24] encode the control flow of the process using Declare constraints. Then, a permutation test is used to identify the constraints for which the difference in confidence between the two event logs is statistically significant. As output, these constraints are translated to natural language. While this approach makes use of statistical tests, they are performed on each Declare constraint separately as opposed to a single, holistic, comparison. This additionally introduces the issue of multiple testing, which raises the risk of false positives. For these reasons, the approach cannot be used for process hypothesis testing. Moreover, the use of Declare constraints limits the approach to the control flow. In [25], a dissimilarity measure between event logs is defined. To this end, a variance matrix is created which encodes differences in directly and eventually follows relations in the event logs. This variance matrix is then used to compute the dissimilarity score. Leemans et al. [3] propose a PHT method that performs a bootstrap test using the EMD and Levenshtein distance to measure dissimilarity in the control flow of the event logs. As such, the setup is similar to the approach presented in this paper, with the key differences being the use of a bootstrap test and the restriction to control flow comparisons. Moreover, their formulation of the test makes the approach asymmetric, and introduces a parameter that can significantly affect the computed p-value. Additionally, our evaluation finds that the approach has a high Type I error rate.

III. Preliminaries

In this section, we introduce the key concepts required to define the proposed approach.

Let Σ_A be a finite set of activities and Σ_D a countably infinite set of event descriptor values used to describe a particular view of the process, for instance, resource or time information. Together, these form an alphabet of events $\Sigma = \Sigma_A \times \Sigma_D$. For an event $e \in \Sigma$, e_a denotes its activity, and e_v denotes its event descriptor value. Then, with Σ^*, we denote the set of possible trace variants in the process.

Definition 1 (Event Log): Let Σ^* be the set of possible trace variants. Then, an event log is a finite multiset of traces $L\colon \Sigma^* \to \mathbb{N}$. With $|L|$, we denote the size of the event log: $\sum_{\sigma \in \Sigma^*} L(\sigma)$.

If we consider only the control flow, we omit the universe of event descriptors, and define $\Sigma = \Sigma_A$.

Stochastic languages can be used to describe a collection of traces by assigning to each trace its relative frequency.

Definition 2 (Stochastic Language): For an alphabet Σ, a *stochastic language* over Σ is a function $f\colon \Sigma^* \to [0, 1]$ with $\sum_{\sigma \in \Sigma^*} f(\sigma) = 1$. For a trace $\sigma \in \Sigma^*$, $f(\sigma)$ denotes its relative frequency in the stochastic language.

An event log L can be translated into a stochastic language over Σ by dividing the absolute trace variant frequency by the total size of the event log. In particular, the event log L induces the stochastic language $f_L(\sigma) = L(\sigma)/|L|$ for all $\sigma \in \Sigma^*$.

Definition 3 (Support of a Stochastic Language): The *support* of a stochastic language f is the set of all items with a nonzero frequency: $S_f = \{\sigma \in \Sigma^* \mid f(\sigma) > 0\}$.

IV. Methods

In this section, we propose the approach for process hypothesis testing. To this end, we first discuss the implications of continuous event descriptors, and how binning can be applied to address their issues. Then, we discuss the weighted Levenshtein distance as a dissimilarity notion between traces. Finally, we introduce the permutation test as a means to compare the event logs based on the EMD and Levenshtein distance to compute a p-value output.

A. Event Descriptor Binning

In principle, any attribute can be used as the event descriptor. However, if its domain is continuous, this can lead to a high level of uniqueness in the extracted traces. As this can cause an explosion in the runtime of the technique and make its application infeasible, we apply binning as a preprocessing step to discretize continuous event descriptors. In particular, we apply binning as proposed by Brockhoff et al. [26] for time-aware concept drift detection. As such, for each activity, we train a binning function on its event descriptor values, which maps event descriptors to a discrete range of k bins. Then, a trace with a real-valued event descriptor is discretized by binning each event descriptor value using the binning function of its associated activity. In particular, K-Means++ clustering [27] is applied with three clusters, and times are binned to the index of the closest centroid. This results in three bins with the semantics of a *fast*, *normal*, and *slow* execution. The chosen number of bins represents a tradeoff between the granularity of considered differences and the runtime of the approach. The output of this step is an event log over the alphabet $\Sigma = \Sigma_A \times \{0, \dots, k-1\}$, for instance, containing the trace $\langle (a, 0), (b, 1), (c, 1), (d, 2) \rangle$, where the execution of a was fast, b and c were normal, and d was relatively slow. Nonetheless, various perspectives of the process can be used as an event descriptor, allowing for the consideration of various process perspectives, depending on the use case.

B. Measuring Dissimilarity

To define a notion of dissimilarity between event logs, we first define a dissimilarity notion between their traces. For this, a trace distance function $\delta\colon \Sigma^* \times \Sigma^* \to \mathbb{R}^{\geq 0}$ is required. Furthermore, δ should yield a distance of 0 if and only if the traces are identical and be symmetric to ensure that the resulting hypothesis test is symmetric. With these requirements in mind, like in [26], we use the *weighted Levenshtein distance*, which can be computed in polynomial time using dynamic programming [28]. Since it allows for assigning dynamic costs for the edit operations, it allows for a more fine-grained edit distance based on the magnitude of difference in the event descriptor.

Definition 4 (Weighted Levenshtein Distance [28]): For a trace $\sigma \in \Sigma^*$, let $hd(\sigma)$ denote the first event in the trace, and $tl(\sigma)$ denote the remaining events after removing the head. Furthermore, let $c_{ins}, c_{del}\colon \Sigma \to \mathbb{R}^{\geq 0}$ and $c_{sub}\colon \Sigma^2 \to \mathbb{R}^{\geq 0}$ be cost functions for inserting and removing an event in a trace and for substituting one event by another, respectively. Then, the weighted Levenshtein distance between two traces $\sigma_1, \sigma_2 \in \Sigma^*$ is defined as:

$$\delta_{lev}(\sigma_1, \sigma_2) = \begin{cases} \sum_{e \in \sigma_2} c_{ins}(e) & \text{if } |\sigma_1| = 0, \\ \sum_{e \in \sigma_1} c_{del}(e) & \text{if } |\sigma_2| = 0, \\ \delta_{lev}(tl(\sigma_1), tl(\sigma_2)) & \text{if } hd(\sigma_1) = hd(\sigma_2), \\ \min \begin{cases} c_{del}(hd(\sigma_1)) + \delta_{lev}(tl(\sigma_1), \sigma_2) \\ c_{ins}(hd(\sigma_2)) + \delta_{lev}(\sigma_1, tl(\sigma_2)) \\ c_{sub}(hd(\sigma_1), hd(\sigma_2)) \\ \qquad + \delta_{lev}(tl(\sigma_1), tl(\sigma_2)) \end{cases} & \text{otherwise} \end{cases}$$

The costs c_{ins}, c_{del}, and c_{sub} must be chosen in such a way that the requirements discussed above are fulfilled. For control flow comparisons, we use the classical Levenshtein distance. This means that a cost of 1 is incurred for insertion and deletion, as well as substitutions of unequal activities. For timed control flow comparisons, we additionally take the binned time differences into account. For this, let k be the number of bins chosen to bin the time dimension and let $\lambda \in [0, 1]$ be a constant determining the weight given to the control flow dimension. Then,

$$c_{ins}(e) = c_{del}(e) = \lambda \cdot 1 + (1 - \lambda) \cdot \frac{e_v}{k - 1}$$

are the costs for insertions and deletions. In the control flow dimension, insertions and deletions, again, incur a unit cost.

In the time dimension, the binned event descriptor value e_v is incurred as cost, scaled to the range $[0, 1]$. The overall cost is then computed as a linear combination of both dimensions using the weight constant λ. The cost for substituting one event, e_1, by another, e_2 is

$$c_{sub}(e_1, e_2) = (1-\lambda) \cdot \frac{|e_{1,v} - e_{2,v}|}{k-1} + \lambda \cdot \begin{cases} 0, \text{ if } e_{1,a} = e_{2,a} \\ 1, \text{ otherwise} \end{cases},$$

where the absolute bin difference, scaled to the range $[0, 1]$ is incurred as time cost. Again, a unit cost is incurred for unequal activities. In contrast to [26], we scale the cost contribution of the time dimension and introduce the weighting constant λ. Using λ, we can control how the edit cost is balanced between the considered dimensions. We recommend a default value of $\lambda = 0.5$, giving an equal weight to both dimensions. Finally, to ensure the comparability of distances between traces of varying lengths, we apply normalization, scaling the Levenshtein distance by the length of the longer trace. In doing so, the Levenshtein distance is scaled to the range $[0, 1]$. In particular, for two traces $\sigma_1, \sigma_2 \in \Sigma^*$, $\overline{\delta}_{lev}(\sigma_1, \sigma_2) = \frac{\delta_{lev}(\sigma_1, \sigma_2)}{\max\{|\sigma_1|, |\sigma_2|\}}$. For example, the weighted Levenshtein distance between the traces $\langle (a, 2), (b, 2), (e, 0), (d, 2) \rangle$ and $\langle (a, 1), (c, 2), (d, 2) \rangle$ is 1.25. First, two substitutions are performed, incurring a cost of 0.25 and 0.5, respectively. Then, as the event $(e, 0)$ does not have a close match in the second trace, a deletion is performed with cost 0.5. As the last events are identical, they are matched with cost 0. Finally, applying normalization yields a normalized Levenshtein distance of $\frac{1.25}{\max\{4,3\}} = 0.3125$.

C. Hypothesis Testing

In the final stage of the process hypothesis testing approach, we perform a hypothesis test, testing the null hypothesis *H_0: The distributions of behavior extracted from the event logs stem from the same distribution*. If we can reasonably reject this null hypothesis, we can conclude that there is a significant difference between the event logs.

1) Permutation Test: The hypothesis test used in this approach is the permutation test [29]. Intuitively, if the null hypothesis holds, so the traces extracted from the event logs l_1 and l_2 stem from the same distribution, then the same holds for their union. Likewise, any two event logs l_1', l_2' created by splitting this union back apart should have similar distributions. This is the core idea of the permutation test we apply. Formally, the dissimilarity between the event logs is first measured using a *test statistic*. Then, traces are randomly shuffled between the event logs and the resulting dissimilarity is measured. This process is repeated N times to compute a multiset of dissimilarities P. The higher N is chosen, the closer the test approximates the case where all permutations were considered. Marozzi et al. [30] recommend $N = 5000$ permutations for a significance level of $\alpha = 0.05$ and $N = 10000$ for $\alpha = 0.01$. Finally, we compare the measured dissimilarities in P to the test statistic t, which was measured between the two event logs in the first step. The p-value is computed as the fraction of values in P that are larger than or equal to t:

$$p = \frac{|[p \in P \mid p \geq t]|}{|P|}. \tag{1}$$

If the computed p-value is sufficiently small, i.e., for some significance level α, we have $p \leq \alpha$, we can reject the null hypothesis and report a statistically significant difference between the event logs.

As the test statistic for the permutation test, we choose the Earth Mover's Distance (EMD) [31] between the stochastic languages of the event logs. Intuitively, the EMD models the dissimilarity between two distributions through piles of earth on the one side, and holes on the other. The work required to move probability mass from a pile to a hole is the distance between those points, and the EMD is the minimum amount of work to flatten all piles by filling all holes.

Definition 5 (Earth Mover's Distance): Let f, g be two stochastic languages over Σ. Furthermore, let $\delta \colon \Sigma^* \times \Sigma^* \to \mathbb{R}^{\geq 0}$ be a distance function between words in the language Σ^*. Then, the Earth Mover's Distance (EMD) between f and g is defined by the optimal transport linear program:

$$\begin{aligned} EMD(f, g) = \min & \sum_{t \in S_f} \sum_{u \in S_g} \delta(t, u) \cdot x_{tu} \\ \text{s.t.} & \sum_{u \in S_g} x_{tu} = f(t) && \forall t \in S_f \\ & \sum_{t \in S_f} x_{tu} = g(u) && \forall u \in S_g \\ & x_{tu} \geq 0 && \forall t \in S_f,\, u \in S_g \end{aligned}$$

The EMD as a dissimilarity measure between event logs using the Levenshtein distance has been proposed for conformance checking [32], concept drift detection [26], and the process hypothesis testing approach of Leemans et al. [3]. Since the EMD is generic regarding the underlying distance notion, it allows us to define custom measures of dissimilarity between trace variants. In particular, this enables the technique to take the *behavioral similarity* of traces into account, in our case, using the Levenshtein distance, making the technique robust to trace variants occurring in only one event log [3].

D. Relation to Existing Work

The existing work by Leemans et al. [3] also uses the Levenshtein distance in combination with the EMD to compare two event logs and compute a p-value output. However, Leemans et al. solely consider the control flow in their comparison, and use a bootstrap test to compute the p-value. The formulation of the test used by Leemans et al., illustrated in Figure 2, operates almost solely on one of the event logs: A distribution of distances is created by computing differences between the first event log and samples taken from it with replacement. Then, the p-value is computed following the same procedure as in the permutation test. Since the bootstrap distribution is solely based on the first event log, the test is asymmetric. Furthermore, the size of the samples has a considerable impact

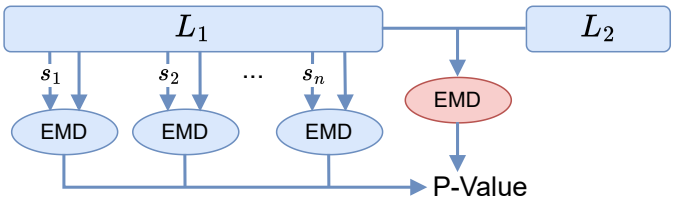

Fig. 2. An overview of the hypothesis test used by Leemans et al. [3].

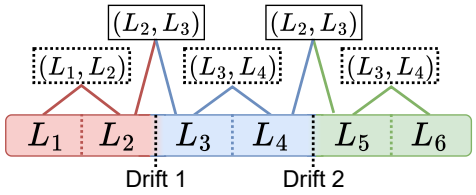

Fig. 3. The procedure for extracting comparison instances from concept drift event logs. Adapted from [7]. Dashed outlines indicate a comparison instance with no difference, while solid outlines indicate a difference.

on the resulting p-value. If the sample size is large, the measured dissimilarity to the source of the sample will be small, and the resulting p-value lower. Conversely, if it is chosen small, the resulting p-value is higher. A parameter that has such an impact on the resulting p-value is undesirable for a hypothesis test. In contrast, our proposed method is symmetric and free of parameters besides the number of permutations.

V. Evaluation

The evaluation of the approach consists of three parts. First, we evaluate its performance in correctly identifying differences in the control flow and compare it to [3]. Then, we investigate whether its Type I error rate behaves as expected. Finally, we evaluate its capability to detect differences in further dimensions in the example of the sojourn time by applying it to a dataset of semisynthetic event logs created from a real-world road traffic fine management process. Implementations of the approach in Python and Rust, as well as the code to reproduce the experiments, are publicly available on GitHub[1,2,3].

A. Control-Flow Detection

To evaluate the control-flow detection capability of the approach, we require a dataset of pairs of event logs with various known ground-truth differences in the control flow. While this kind of dataset does not exist for process comparison or process hypothesis testing techniques, there are a number of such datasets that are used in the evaluation of concept drift detection techniques [33]. These datasets provide synthetic event logs that contain sudden concept drifts in various aspects of the control flow and form an ideal scenario for evaluating the different kinds of control flow differences that the approach is capable of detecting. Brockhoff et al. [7] adapt the event log datasets used in [33] to create comparison instances that can be used for the evaluation of process comparison techniques. The procedure for extracting comparison instances is shown in Figure 3. Positive comparison instances contain a difference and are created around concept drifts, while negative comparison instances are built using sub-logs from the same stable period of an event log and contain no differences. Applying this procedure to the event logs in [34]–[36], we create 508 comparison instances, of which 198 contain a difference and 310 do not. Including the event logs with added noise, this number increases to 1968 comparison instances, where 758 contain a difference and 1210 do not. For an overview of types of control flow differences contained in these event logs, we refer the reader to the respective papers [34]–[36] as well as the definitions of change patterns proposed in [37].

1) Evaluation Measures: Applying the technique to an event log pair with a known ground-truth gives a detection which can be classified as either a *true positive* (TP), *false positive* (FP), *true negative* (TN), or *false negative* (FN). Using these, we can compute the *statistical power* (or recall) of the approach as the fraction of instances containing a difference where the null hypothesis was rejected. It is computed as

$$power = \frac{TP}{TP + FN}. \tag{2}$$

The Type I error rate (or false positive rate) is the fraction of negative instances where the null hypothesis was incorrectly rejected. It is computed as

$$T1\text{-}ER = \frac{FP}{FP + TN}. \tag{3}$$

The chosen significance level, α, sets an upper limit on the acceptable Type I error rate [38]. As such, we do not expect a Type I error rate of 0, but rather $T1\text{-}ER \leq \alpha$. Furthermore, we consider the accuracy and F1-score.

2) Parameter Settings: The only free parameters of the proposed approach are the number of permutations to perform, N, and the significance level, α. For the significance level, we choose the commonly used value $\alpha = 0.05$ [39]. As the number of permutations, we choose $N = 10000$. As discussed in Section IV, a recommended value for $\alpha = 0.05$ is $N = 5000$, so our choice should be sufficient. For the approach of Leemans et al., we use the parameters used by the authors in their evaluation, i.e., $N = 10000$ and $resample\_size = |L_1|$.

3) Results: Table II shows the evaluation measures of the approaches after applying them to the control flow evaluation dataset. Overall, the proposed approach is indeed capable of detecting differences in many instances, achieving a statistical

TABLE II
Performance on the control-flow event logs without noise

Approach	Accuracy	Power	T1-ER	F1-Score
Bootstrap Test	0.52	0.96	0.76	0.61
Permutation Test	0.83	0.91	0.22	0.81

[1]https://www.github.com/cpitsch/pcomp
[2]https://www.github.com/cpitsch/pcomp-rs
[3]https://www.github.com/cpitsch/pcomp-experiments

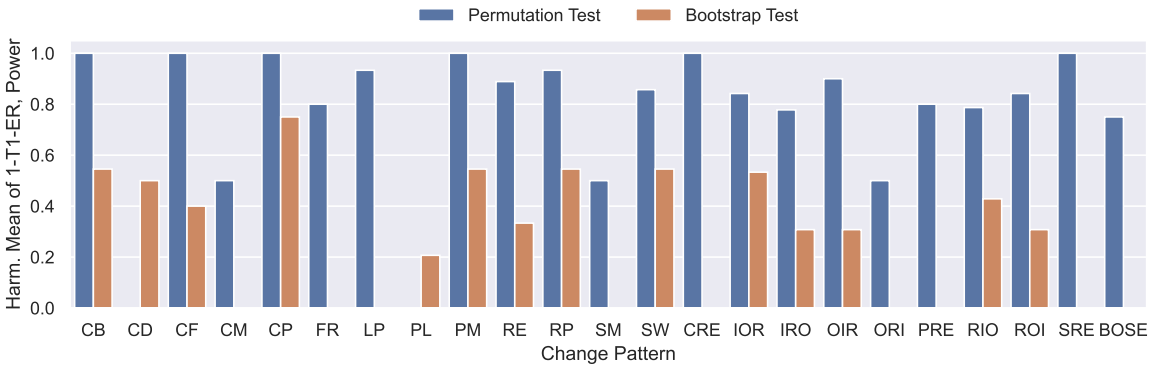

Fig. 4. Harmonic mean of power and $1 - T1\text{-}ER$ for each change pattern. A higher number indicates better performance. Missing values indicate change patterns where either no differences were detected, or $T1\text{-}ER = 1$. Descriptions of the change patterns are found in [37] and the respective papers [34]–[36].

power of approximately 0.91, and a Type I error rate of 0.22. While the Type I error rate is higher than the expected value of 0.05, the permutation test approach performs better than the bootstrap-based approach, which has a Type I error rate of 0.76, incorrectly classifying 76% of the negative comparison instances as having a difference. The proposed approach also achieves a considerably higher accuracy and F1-score than the bootstrap-based approach. Figure 4 shows a breakdown of the performance into the individual change patterns, measured by the harmonic mean of statistical power and the complement of the Type I error rate. For instance, the *ParallelToSequence* (`PL`) change pattern makes a parallel portion of the process sequential, and vice versa. In-depth definitions of the covered change patterns can be found in [34]–[37]. For simplicity, the event log of Bose et al. [34] is shown as a separate change pattern. Overall, the permutation test achieves a high score on many change patterns, and consistently outperforms the bootstrap-based approach. Notably, the permutation test approach does not detect any differences in the comparison instances based on the event logs of Ceravolo et al. [36] for the change patterns *Synchronize* (`CD`) and *ParallelToSequence* (`PL`). Upon manual inspection, these event logs do not seem to contain any concept drift: Both change patterns involve concurrency. However, in both cases, the parallel activities of the process only occurred in one order, meaning that the change patterns had no effect.

In Tables III and IV, we show the performance of the approaches on the synthetic event logs containing noise. Since the authors in [35] and [36] use different notions of noise, the analysis is performed on each dataset separately. On the dataset of Ceravolo et al., the permutation test approach is more robust than the bootstrap test approach, with the Type I error rate remaining approximately constant over all noise levels, achieving a Type I error rate of 0.03 on the highest noise level. In contrast, the Type I error rate of the bootstrap test approach increases from 0.51 without noise to 0.75 at the highest level. On the data of Ostovar et al., the permutation test is less robust, with its Type I error rate increasing from 0.39 to 0.43, while the bootstrap test has a Type I error rate of 1 on all

TABLE III
PERFORMANCE ON THE EVENT LOGS OF [36] CONTAINING NOISE.

Noise	Approach	Accuracy	Power	T1-ER	F1-Score
0%	Bootstrap Test	0.65	0.91	0.51	0.66
	Permutation Test	0.92	0.84	0.04	0.88
5%	Bootstrap Test	0.55	0.89	0.65	0.60
	Permutation Test	0.93	0.84	0.03	0.89
10%	Bootstrap Test	0.57	0.96	0.67	0.62
	Permutation Test	0.93	0.87	0.03	0.91
15%	Bootstrap Test	0.51	0.98	0.77	0.60
	Permutation Test	0.92	0.84	0.04	0.88
20%	Bootstrap Test	0.53	0.98	0.75	0.61
	Permutation Test	0.93	0.87	0.03	0.91

TABLE IV
PERFORMANCE ON THE EVENT LOGS OF [35] CONTAINING NOISE.

Noise	Approach	Accuracy	Power	T1-ER	F1-Score
0%	Bootstrap Test	0.40	1.00	1.00	0.57
	Permutation Test	0.75	0.96	0.39	0.76
2%	Bootstrap Test	0.40	1.00	1.00	0.57
	Permutation Test	0.77	0.98	0.37	0.77
10%	Bootstrap Test	0.40	1.00	1.00	0.57
	Permutation Test	0.74	1.00	0.43	0.76

noise levels. Together with its power of 1 on all noise levels, this means that the bootstrap test approach detects a difference in every instance. In this regard, the permutation test approach performs better on the data of Ostovar et al. Nonetheless, its Type I error rate is considerably higher than the expected value of 0.05, even without noise. This is likely because these event logs are rather short, but contain very complex and long traces, and do not represent the underlying process well. This creates high uniqueness even in stable periods of the event logs, which causes differences to be detected.

B. Type I Error Rate

As mentioned in Section V-A1, the expected Type I error rate is upper-bounded by the significance level, α. To investigate this relationship, we investigate the Type I error rate as a function of the chosen significance level. To this end, we create a dataset of comparison instances with *no difference* by randomly splitting the real-world Road Traffic Fine

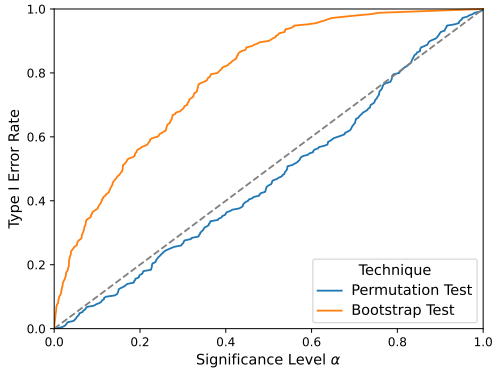

Fig. 5. The Type I error rate of the approaches as a function of the significance level. The expected upper bound of $T1\text{-}ER = \alpha$ is shown as a gray dashed line.

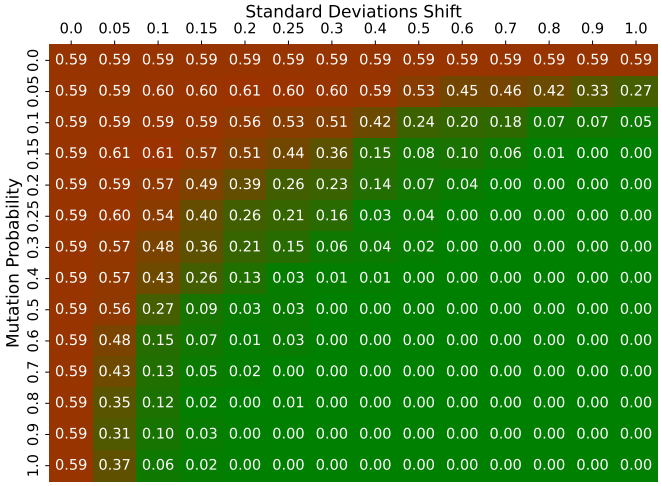

Fig. 6. The mean p-value for each parameter setting of the sensitivity analysis.

Management (RTFM) event log [40] into halves 250 times. Since, to our knowledge, the RTFM event log does not contain any known concept drifts, these splits should exhibit the same behavior, and thus, be classified as having no difference. The result of applying both control-flow PHT techniques to this dataset is shown in Figure 5. The permutation test approach closely follows the expected upper bound of $T1\text{-}ER = \alpha$, while the bootstrap test has a much higher Type I error rate than the expected value. Overall, the permutation test approach behaves as expected regarding its Type I error rate.

C. Timed Control Flow Sensitivity

Finally, we analyze the sensitivity of the approach to differences in the timed control flow. Since no synthetic datasets exist that contain time differences in varying severity, we create a dataset of *semisynthetic* event logs by injecting mutations into the real-world RTFM event log [40]. In doing so, we can control the severity and frequency of differences while having a realistic level of noise in the event data, which is inherited from the real-world event log. To this end, for every "Send Fine" event in the event log, with a certain probability, its sojourn time is increased by a factor of its standard deviation. To avoid control flow changes, the following events are shifted by the same amount. In this way, it is guaranteed that the differences that are induced through this procedure are solely in the time dimension, allowing for a fine-grained analysis of the time-difference detection capability. As such, there are two parameters to the mutation procedure: the probability of injecting a mutation per matching event, and the severity of the mutation, i.e., the factor of the standard deviation by which to increase the sojourn time. We take both of these values from the set $\{0.0, 0.05, 0.1, \ldots, 0.25, 0.3, 0.4, \ldots, 1.0\}$. Furthermore, to adjust for the randomness in the mutations, we run the experiment for five seeds, resulting in a total of $5 \cdot 14^2 = 980$ comparison instances.

1) Results: The results of applying the proposed approach to the semisynthetic dataset are shown in Figure 6. Here, the mean p-value, aggregated over all five seeds, is shown for each mutation parameter setting. As expected, if either the mutation probability or severity is zero, no mutation is injected into the event log, and thus, no difference is detected with a mean p-value of 0.59. By increasing the probability and severity of mutations, the magnitude of difference increases, and the mean p-value decreases. For instance, for a probability of 0.4 and a severity of 0.25 standard deviations, the mean p-value is 0.03, on average, detecting a difference for a significance level of 0.05. For higher values of probability and severity, the mean p-value consistently reaches approximately 0.0, indicating that the approach is capable of detecting these differences.

VI. Conclusion

In this paper, we have proposed an approach for hypothesis testing in process mining for various process dimensions. To this end, sequences of activities and event descriptor values are first extracted from event logs. Then, a permutation test is performed using the Earth Mover's Distance (EMD) as test statistic, and the weighted Levenshtein distance to measure dissimilarity between trace variants. While the approach has been implemented for control flow and timed control flow comparisons, it is also applicable to other numerical dimensions. Furthermore, by defining a different cost function for the EMD, various kinds of constructs can be extracted from the event logs and compared. The evaluation on synthetic control-flow concept drift datasets shows that the proposed approach performs well in detecting various kinds of control flow differences, while performing better than existing approaches. The investigation of the Type I error rate shows that the approach indeed behaves as expected in terms of the Type I error rate. Finally, the evaluation on a semisynthetic dataset containing time differences shows that the approach is also capable of detecting differences in the time dimension.

While the approach performs well in detecting differences, it does not explain their nature. Therefore, in future work, we plan to investigate techniques to leverage the flow matrix computed in the EMD computation to expose diagnostics explaining the nature of the detected difference. Furthermore,

the approach uses binning to deal with continuous dimensions. However, this has the limitation that it reduces the granularity of considered differences. For this reason, further research could include techniques to work directly with numerical values, as opposed to binning them. This would allow for a more fine-grained comparison, considering more accurate time differences. Moreover, in this paper, we have only discussed the instantiation of the Levenshtein distance for numerical event descriptors. In this regard, we plan to explore further trace representations and dissimilarity measures that can be used to capture further perspectives of the process. Since the number of permutations N plays a significant role in the runtime of the approach, we also plan to analyze the impact of different choices for this parameter on the computed p-value. Furthermore, *e-values* have been proposed as an alternative to p-values that have better interpretability by quantifying evidence against the null hypothesis [41]. In this regard, future work could include the use of e-values in place of p-values. Finally, we intend to apply the technique to real-world medical data, since process hypothesis testing techniques can be instrumental in investigating treatment processes for disparities.

REFERENCES

[1] W. M. P. van der Aalst, *Process Mining - Data Science in Action, Second Edition*. Springer, 2016.

[2] Centers for Medicare and Medicaid Services (CMS), "HHS finalizes rule to strengthen medicare, improve access to affordable prescription drug coverage, and hold private insurance companies accountable to delivering quality health care for america's seniors and people with disabilities," "https://www.hhs.gov/about/news/2023/04/05/hhs-finalizes-rule-strengthen-medicare-improve-access-affordable-prescription-drug-coverage-hold-private-insurance-companies-accountable.html", 2023, [Online; accessed 2024-11-05].

[3] S. J. J. Leemans, J. M. McGree, A. Polyvyanyy, and A. H. M. ter Hofstede, "Statistical tests and association measures for business processes," *IEEE Transactions on Knowledge and Data Engineering*, vol. 35, no. 7, 2023.

[4] N. R. T. P. van Beest, M. Dumas, L. García-Bañuelos, and M. L. Rosa, "Log delta analysis: Interpretable differencing of business process event logs," in *BPM*, 2015.

[5] A. Jalali, P. Johannesson, E. Perjons, Y. Askfors, A. R. Kalladj, T. Shemeikka, and A. Vég, "dfgcompare: a library to support process variant analysis through markov models," *BMC Medical Informatics and Decision Making*, vol. 21, no. 1, 2021.

[6] M. Vidgof, D. Djurica, S. Bala, and J. Mendling, "Interactive log-delta analysis using multi-range filtering," *Software Systems and Modeling*, vol. 21, no. 3, 2022.

[7] T. Brockhoff, M. S. Uysal, and W. M. P. van der Aalst, "Process comparison based on selection-projection structures," in *CAiSE*, 2024.

[8] J. C. A. M. Buijs and H. A. Reijers, "Comparing business process variants using models and event logs," in *BPMDS/EMMSAD*, 2014.

[9] A. Rozinat and W. M. P. van der Aalst, "Conformance checking of processes based on monitoring real behavior," *Information Systems*, vol. 33, no. 1, 2008.

[10] A. Adriansyah, "Aligning observed and modeled behavior," Ph.D. dissertation, Mathematics and Computer Science, 2014.

[11] B. Cao, J. Wang, J. Fan, S. Deng, J. Yang, W. Zhao, J. Yin, and M. Zhou, "Prodiff: A process difference detection method based on hierarchical decomposition," *IEEE Transactions on Services Computing*, vol. 15, no. 1, 2022.

[12] J. Wang, B. Cao, X. Zheng, D. Tan, and J. Fan, "Detecting difference between process models using edge network," *IEEE Access*, vol. 7, 2019.

[13] A. Pini, R. Brown, and M. T. Wynn, "Process visualization techniques for multi-perspective process comparisons," in *AP-BPM*, 2015.

[14] N. P. Ballambettu, M. A. Suresh, and R. P. J. C. Bose, "Analyzing process variants to understand differences in key performance indices," in *CAiSE*, 2017.

[15] M. T. Wynn, E. Poppe, J. Xu, A. H. M. ter Hofstede, R. Brown, A. Pini, and W. M. P. van der Aalst, "ProcessProfiler3D: A visualisation framework for log-based process performance comparison," *Decision Support Systems*, vol. 100, 2017.

[16] J. Gulden, "Visually comparing process dynamics with rhythm-eye views," in *BPM Workshops*, 2016.

[17] E. Toraman, "Visualizing business process deviance with timeline diagrams," Master's thesis, University of Tartu, 2018.

[18] H. Nguyen, M. Dumas, M. L. Rosa, and A. H. M. ter Hofstede, "Multi-perspective comparison of business process variants based on event logs," in *ER*, 2018.

[19] J. Cremerius, H. Patzlaff, V. X. Rahn, and H. Leopold, "Data-based process variant analysis," in *HICSS*, 2023.

[20] F. Taymouri, M. L. Rosa, and J. Carmona, "Business process variant analysis based on mutual fingerprints of event logs," in *CAiSE*, 2020.

[21] A. Bolt, M. de Leoni, and W. M. P. van der Aalst, "A visual approach to spot statistically-significant differences in event logs based on process metrics," in *CAiSE*, 2016.

[22] ——, "Process variant comparison: Using event logs to detect differences in behavior and business rules," *Information Systems*, vol. 74, no. Part, 2018.

[23] T. Brockhoff, M. N. Gose, M. S. Uysal, and W. M. P. van der Aalst, "Process comparison using petri net decomposition," in *PETRI NETS*, 2024.

[24] A. Cecconi, A. Augusto, and C. D. Ciccio, "Detection of statistically significant differences between process variants through declarative rules," in *BPM Forum*. Springer, 2021.

[25] F. Corradini, C. Luciani, A. Morichetta, M. Piangerelli, and A. Polini, "Tlv-dissγ: A dissimilarity measure for public administration process logs," in *EGOV*, 2021.

[26] T. Brockhoff, M. S. Uysal, and W. M. P. van der Aalst, "Time-aware concept drift detection using the earth mover's distance," in *ICPM*, 2020.

[27] D. Arthur and S. Vassilvitskii, "k-means++: the advantages of careful seeding," in *SODA*, 2007.

[28] R. A. Wagner and M. J. Fischer, "The string-to-string correction problem," *Journal of the ACM*, vol. 21, no. 1, 1974.

[29] E. J. G. Pitman, "Significance tests which may be applied to samples from any populations," *Supplement to the Journal of the Royal Statistical Society*, vol. 4, no. 1, 1937.

[30] M. Marozzi, "Some remarks about the number of permutations one should consider to perform a permutation test," *Statistica*, vol. 1, 01 2004.

[31] Y. Rubner, C. Tomasi, and L. J. Guibas, "A metric for distributions with applications to image databases," in *ICCV*, 1998.

[32] S. J. J. Leemans, W. M. P. van der Aalst, T. Brockhoff, and A. Polyvyanyy, "Stochastic process mining: Earth movers' stochastic conformance," *Information Systems*, vol. 102, 2021.

[33] J. N. Adams, C. Pitsch, T. Brockhoff, and W. M. P. van der Aalst, "An experimental evaluation of process concept drift detection," *Proceedings of the VLDB Endowment*, vol. 16, no. 8, 2023.

[34] R. P. J. C. Bose, W. M. P. van der Aalst, I. Zliobaite, and M. Pechenizkiy, "Dealing with concept drifts in process mining," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 25, no. 1, 2014.

[35] A. Ostovar, S. J. J. Leemans, and M. L. Rosa, "Robust drift characterization from event streams of business processes," *ACM Transactions on Knowledge Discovery from Data*, vol. 14, no. 3, 2020.

[36] P. Ceravolo, G. M. Tavares, S. B. Junior, and E. Damiani, "Evaluation goals for online process mining: A concept drift perspective," *IEEE Transactions on Services Computing*, vol. 15, no. 4, 2022.

[37] B. Weber, M. Reichert, and S. Rinderle-Ma, "Change patterns and change support features - enhancing flexibility in process-aware information systems," *Data & Knowledge Engineering*, vol. 66, no. 3, 2008.

[38] P. Pollard and J. T. Richardson, "On the probability of making type I errors." *Psychological Bulletin*, vol. 102, no. 1, Jul. 1987.

[39] D. R. Anderson, K. P. Burnham, and W. L. Thompson, "Null hypothesis testing: Problems, prevalence, and an alternative," *The Journal of Wildlife Management*, vol. 64, no. 4, 2000.

[40] M. M. de Leoni and F. Mannhardt, "Road traffic fine management process," 2015. [Online]. Available: https://data.4tu.nl/articles/dataset/Road_Traffic_Fine_Management_Process/12683249/1

[41] P. Grünwald, R. de Heide, and W. M. Koolen, "Safe testing," in *ITA*, 2020.