

# Fairness in Prescriptive Process Analytics: Integrating Fairness Constraints into the Learning Phase

Fabrizio De Santis<sup>1\*</sup>, Alessandro Padella<sup>2\*</sup>, Gyunam Park<sup>3</sup>, Francesco  
Zanichelli<sup>1</sup>, and Massimiliano de Leoni<sup>2</sup>

<sup>1</sup> University of Parma, Parma, Italy

{fabrizio.desantis, francesco.zanichelli}@unipr.it

<sup>2</sup> University of Padua, Italy

alessandro.padella@unipd.it deleoni@math.unipd.it

<sup>3</sup> Eindhoven University of Technology, Eindhoven, Netherlands  
g.park@tue.nl

**Abstract.** Prescriptive Process Analytics is a core field in business process management that provides automated activity recommendations to improve Key Performance Indicators in business processes. Because such recommendations can directly affect individuals involved in the process, ensuring fairness is essential. Recent work has begun to address fairness in process monitoring by enforcing group-level parity across demographic groups at the predictive stage. However, these approaches leave the prescriptive phase entirely unexamined and rely solely on group fairness notions, failing to account for individual-level counterfactual fairness (i.e., the requirement that a recommendation should not change when sensitive attributes are altered). To address this gap, we propose to enforce counterfactual fairness directly in prescriptive process monitoring. We develop a novel neuro-symbolic framework based on Logic Tensor Networks that encodes counterfactual fairness as a logical axiom over predictions, seamlessly integrated into the learning process. We evaluate our approach on real-world event logs and find that it enforces counterfactual fairness at the prediction level and improves recommendation-level counterfactual consistency under the considered policy, while maintaining competitive predictive and prescriptive performance, establishing the first fairness-aware framework for Prescriptive Process Analytics.

**Keywords:** Fairness · Prescriptive Process Analytics · Predictive Process Monitoring · Logic Tensor Networks.

## 1 Introduction

Process Mining is a discipline that bridges data science and business process management, extracting knowledge from event logs recorded by information systems to analyze, understand, and improve business processes. Within this field, Prescriptive Process Analytics (PrPA) aims to focus on recommending actions to mitigate “risky” executions that are predicted to lead to unsatisfactory outcomes, such as high costs or poor customer satisfaction.

---

\* F. De Santis and A. Padella—Equal contribution.

Since such recommendations can directly affect the individuals involved in a process, ensuring that they are not driven by discriminatory criteria is essential. This paper addresses the problem of frameworks for PrPA that provide unfair recommendations, namely recommendations that are based on characteristics that discriminate in a form that is unacceptable from a legal and/or ethical point of view. For instance, in a system managing loan applications, an applicant might be requested for additional checks on the basis of their gender. Pohl et al. indicate monitoring, detecting and rectifying biased patterns to be the most significant challenge in Discrimination-Aware Process Mining [19]. Process characteristics are modeled as process attributes, where sensitive attributes are those that cannot be used to provide recommendations.

Despite growing interest in fairness in machine learning, existing work on fairness in Process Mining has focused exclusively on the predictive stage, aiming to ensure that sensitive attributes do not influence outcome predictions, adopting group fairness notions. This body of research work can certainly be adapted for PrPA, but they adopt the group fairness notions, which require statistical parity across demographic groups at the population level. Prescriptive settings also require asking whether the same running instance would receive a different recommendation under a counterfactual change of the sensitive attribute. This is captured by counterfactual fairness [13].

The importance of individual-level fairness can be expressed through an example of a process at a hypothetical hospital’s emergency department. After assessing the symptoms and analyzing the lab results, the system recommends the next step in the medical process. Suppose that initially these recommendations may be unfair, and old patients are sent to cardiology, while young patients are sent elsewhere. In this scenario, when the daily “quota” for old-patient cardiology is reached, additional old patients at triage may be sent to other departments, regardless of the symptoms, and results: this would ensure group fairness. Although this may improve aggregate statistics, it could be dangerous for individual patients. Counterfactual fairness captures this individual-level concern: for a given patient, if we image a counterfactual version of the same individual where age is different, while symptoms and lab results remain unchanged, the recommendation should remain the same. This does not mean that counterfactual fairness is generally superior to group fairness. Group fairness remains essential for auditing and regulatory reporting, where aggregate disparities must be assessed. For example, a loan system may satisfy group-level parity by balancing individually unfair decisions across nationalities, still treating single applicants unfairly. Conversely, counterfactual fairness does not guarantee group parity.

This paper focuses on this individual-level layer and proposes the first fairness-aware framework for PrPA that enforces counterfactual fairness. In particular, it builds on Logic Tensor Networks (LTNs), a neuro-symbolic framework that maximizes the satisfaction of a set of logical formulas while fitting the observed data [2]. We define regression axioms for accurate KPI prediction and counterfactual-fairness axioms requiring prediction invariance under sensitive-attribute interventions, using counterfactuals identified through causal discovery and inference. Our fairness-aware framework has been evaluated on real-world logs and multiple sensitive attributes. Results show that it increases fairness with the same recommendation effectiveness as a framework not tailored for it.

The remainder of this paper is structured as follows: Sect. 2 discusses the previous literature pertaining to fairness in Process Mining and PrPA. Sect. 3 introduces the background concepts, and Sect. 4 proposes a framework that enables individually fair prescriptions. The case studies, the evaluation setup, and the results are described in Sect. 5, while Sect. 6 discusses the results and their limitations. Finally, Sect. 7 concludes the paper by summarizing our contribution and discussing future directions.

## 2 Related Work

Fairness has been identified as one of the four main challenges of Responsible Process Mining [16]. In parallel, PrPA has gained increasing attention in recent years [12]. Early approaches focused on recommending the next activity to execute by leveraging basic forms of counterfactual reasoning [8], with the primary goal of improving business process performance. Subsequently, in parallel with the rise of Predictive Process Monitoring (PPM) methods [7], and given that accurate predictions do not necessarily imply effective recommendations [6], the focus of PrPA expanded toward anticipating which ongoing process instances might lead to unfavorable KPIs [22]. The adoption of deep-learning techniques has enabled the enhancement of prescriptive capabilities, leading the research to explore the integration of causal inference [3] and proposing white-box approaches [20]. Recent advancements have focused on approaches that optimize a given KPI through recommending the optimal resource allocation [17], or jointly recommending both the activity to perform and the resource responsible for its execution [18], also integrating Large Language Models to enhance the interpretability of recommendations [11].

Despite the progress in PrPA, fairness within this field has received significantly less attention than in PPM where group fairness prediction approaches have been developed [14,15]. This is further confirmed in [21], a recent survey on fairness in PPM, which reports the evaluation of group fairness prediction methods and identifies fair prescriptive process analysis as an open direction for future work. Although counterfactual fairness has not yet been studied in PrPA, Heilmann et al. [9] proposed a neuro-symbolic approach for enforcing counterfactual fairness by encoding fairness constraints over model predictions using LTNs. Their work provides the fairness-aware learning mechanism that we build upon. However, their proposal is not directly applicable in our settings, where our framework provides recommendations by looking at the sequences of actions, namely the execution traces, that were executed for individual process instances. Indeed, we generate a counterfactual trace for each trace, thus ultimately creating a counterfactual event log. This step is novel, if compared with Heilmann et al. [9], and it is crucial to enable fairness-aware prescriptive process analytics.

## 3 Preliminaries

Consider a loan application process. For a running process instance, the information system records the sequence of activities, such as *Application submitted*, *Documents requested*, *Documents verified*, *Risk assessment*, *Manual review*, *Additional document request*, *Fast-track assessment*, and *Application approved* or

*Application accepted.* In addition to the control-flow, it also records the timestamps of the activities, the resources executing them, and they can also be associated with attributes such as *employment\_type*, *requested\_amount*, and *gender*. The goal of PrPA is to recommend the activity to perform as next to improve a pre-defined KPI. Suppose that, after a sequence of activities such as Application submitted, Documents requested, and Documents verified, a PrPA framework recommends the next activity to minimize the remaining processing time between the eligible next activities that are Manual review, Fast-track assessment, and Additional document request. For a male applicant, the system predicts remaining times of 6, 4, and 9 days, respectively, and recommends Fast-track assessment. Now consider the counterfactual version of the same case, where the applicant is female and gender-dependent attributes are updated accordingly. If the predicted remaining times become 5, 7, and 8 days, the recommendation changes to Manual review. Counterfactual fairness captures this individual-level concern: changing gender alone should not change the recommendation for the same case. The rest of this section outlines the core concepts behind our PrPA framework.

### 3.1 Prescriptive Process Analytics

An event log  $\mathcal{L}$  is a collection traces, each describing one process instance as an ordered sequence of events  $\sigma = \langle e_1, e_2, \dots, e_n \rangle$  sharing the same case identifier  $c$ , where every event  $e = (a, c, t, attr_1, \dots, attr_m)$  captures the executed activity  $a$ , its timestamp  $t$ , and optional payload attributes. A running trace is represented by a prefix  $p = (\sigma, k)$  that collects the first  $k$  events of  $\sigma$ . We denote the set of all prefixes extracted from an event log  $\mathcal{L}$  as  $\mathcal{P}$ , and as  $\mathcal{K} : \mathcal{P} \rightarrow \mathbb{R}$  a function that maps each prefix  $p$  to its own KPI.

Given a running trace  $p$ , the goal of a PrPA framework is to recommend the optimal next activity to execute. However, this activity has to be practically executable from the domain viewpoint. To ensure it, we leverage a *transition systems* [1]. Given an event log  $\mathcal{L}$ , each prefix in  $\mathcal{P}$  is mapped to an abstract state based on the last few activities, leveraging a *trace-to-instance* encoding function. Given a prefix  $p = (\sigma, k)$  in this paper it is formulated as  $\rho_a(p) = \langle a_1, \dots, a_k \rangle$ . Given this state, the transition system, built from  $\mathcal{L}$ , identifies which activities are eligible as next activities to recommend.

In parallel with the transition systems, a PPM model  $f : \mathcal{P} \rightarrow \mathbb{R}$  learns to estimate the KPI from a running trace. For a prefix  $p$ , the model  $f$  tries to forecast the value of  $\mathcal{K}(p)$ . Then, once the set of eligible next activities for  $p$  has been extracted from the transition system, the same PPM model is used to evaluate the expected KPI of the valid next activities, recommending the activity that is predicted to lead to the most favourable KPI.

### 3.2 Counterfactual Fairness

The PPM model  $f$  may learn to associate sensitive attributes with process outcomes, producing different predictions for individuals who differ only in these attributes. This form of algorithmic discrimination can propagate from predictions to recommendations, leading to unfair treatment of specific individuals.

Existing approaches to fairness in PPM adopt group fairness notions (cf. Sect. 2). However, it provides no guarantee about the recommendations provided to specific individuals and does not address the question: *would this running process execution  $p$  have received a different prediction/prescription if its sensitive attribute had been different?* Counterfactual fairness [13] addresses this limitation by stipulating that a decision or treatment must remain invariant when the sensitive attribute is intervened upon, provided that all other variables are propagated in accordance with the underlying causal mechanisms of the model.

Counterfactual fairness in Process Mining requires that for a running prefix  $p$  and its counterfactual version  $p^{cf}$ , the PPM predictor  $f$  predicts the same KPI, while the PrPA recommender prescribes the same activity for both cases. In the loan example, suppose the prefix is  $\langle \textit{Application submitted}, \textit{Documents requested}, \textit{Documents verified} \rangle$  for a male applicant, and the transition system returns as eligible next activities *Manual review*, *Additional document request*, and *Fast-track assessment*, with predicted remaining times of 6, 9, and 4 days respectively. The system recommends *Fast-track assessment*. In the counterfactual version where the applicant is female and causally affected attributes are updated accordingly, the predicted remaining times become 5, 8, and 7 days, and the recommendation changes to *Manual review*. This illustrates the fairness problem addressed in this work: changing the applicant’s gender from male to female should not lead to a different recommendation.

The main challenge is that meaningful counterfactuals cannot be obtained by simply flipping the sensitive attribute, as doing so may also affect causally downstream attributes. Counterfactual fairness therefore requires a causal model.

A Structural Causal Model is a tuple  $\mathcal{M} = (U, V, F)$  where  $U$  is a set of exogenous variables representing background factors that are not caused by any variable in the model, such as personal or institutional circumstances not recorded in the event log;  $V$  is a set of endogenous variables, whose values are ultimately determined by the exogenous variables, such as gender, employment type, and requested loan amount;  $F$  is a set of structural equations that determine the value of each endogenous variable [13]. For instance, one equation may specify how employment type depends on gender and other background factors, while another may specify how the requested amount depends on employment type. An SCM provides a formal description of how variables of a system are generated and how they influence each other. In a counterfactual query, we ask what the rest of the case would look like if a variable were changed, while the causal mechanisms of the system remain the same. In the loan example, changing gender may affect downstream attributes such as employment type, but not the requested loan amount. Intervening on gender would also update the employment type, while leaving the requested amount unchanged. An SCM specifies how these downstream variables should be recomputed after this intervention.

## 4 Counterfactual Fairness for Prescriptive Process Analytics

Fig. 1 provides an overview of the proposed approach. Before training, the event log is used to construct a transition system encoding valid next activities for any process state, and to generate a counterfactual event log in which each trace is

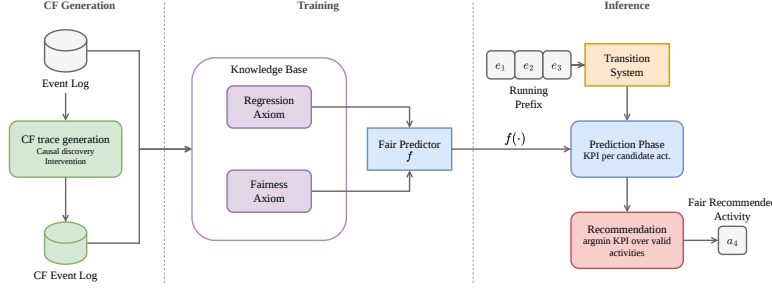


Fig. 1: Overview of the proposed framework. Counterfactual traces are generated from the event log via causal discovery and intervention, and used to ground a counterfactual fairness axiom alongside a regression axiom in the knowledge base; joint optimization of the two yields a fair predictor. At inference time, the model predicts the KPI for each candidate activity from the transition system, and the activity with the best predicted KPI is recommended.

paired with counterfactual counterparts obtained by intervening on the sensitive attribute. Original and counterfactual prefix pairs are then used to ground two logical axioms: a regression axiom for accurate KPI prediction, and a counterfactual fairness axiom requiring invariant predictions across counterfactual pairs. Joint optimization of the two axioms yields a fair predictor. At inference time, the transition system determines the candidate next activities for a running process prefix, the model predicts the KPI for each, and the activity with the best predicted KPI is recommended.

#### 4.1 Counterfactual Trace Generation

Enforcing counterfactual fairness in Process Mining requires, for each trace  $\sigma \in \mathcal{L}$ , one or more corresponding counterfactual traces in which the sensitive attribute has been altered while everything else is updated consistently with the underlying causal structure. Generate such counterfactual traces from an event log  $\mathcal{L}$ , requires knowledge of the causal structure under the data-generation process, encoded as an SCM. Since it is rarely known in advance, we employ a causal discovery algorithm to infer it from the data. Specifically, we use the Fast Causal Inference (FCI) [4], a constraint-based causal discovery algorithm that takes as input a dataset of variables and returns a Partial Ancestral Graph (PAG): a graphical representation encoding the set of all causal structures consistent with the data, including the possible presence of latent confounders.

We adopt two assumptions that constrain which parts of a trace need to be modified when intervening on the sensitive attribute.

*Control-flow independence.* The transition system constructed from the event log (cf. Sect. 3.1) is assumed to be independent of the sensitive attribute, meaning that the set of possible next activities for a given prefix is the same regardless of the demographic characteristics of the process instance. Therefore, we assume that the sequence of activities executed is determined by organizational rules and process logic rather than individual demographic characteristics. The sen-

sitive attribute may, however, influence which activity is recommended by the predictor, since the predicted KPIs can differ across counterfactual pairs.

*Case-level causal dependence.* We assume that only case-level attributes can causally depend on the sensitive attribute. This assumption is grounded in a principled distinction between *stable* and *unstable* attributes. Case-level attributes are stable, meaning that they are defined once at the case level and do not change across events. This makes them suitable anchors for a causal model whose structure is assumed fixed across a trace. Event-level attributes, on the other hand, can take different values and are therefore considered unstable. Including such unstable attributes in the causal model would destabilize the causal structure itself, since the causal relationships between the variables would have to be recalculated for each event.

FCI is applied to the tabular case-level representation extracted from the data, where each row represents a case with its corresponding case-level attributes. From the resulting PAG, we extract causal dependencies between the sensitive attribute and the case-level attributes, identifying which attributes are causally downstream of the sensitive attribute and must be updated when generating counterfactual traces. To reduce the risk of propagating uncertain causal relations, we only recompute case-level attributes for which the discovered PAG supports a directed or possibly directed path from the sensitive attribute. All other attributes are kept fixed. In the presence of latent confounders, indicated by bi-directed edges in the PAG returned by FCI, the affected attributes are treated as causally independent of the sensitive attribute and therefore left unchanged. In the loan example, the sensitive attribute is gender and the case-level attributes include employment type and requested amount. If FCI returns a directed edge from gender to employment type, reflecting historical bias in employment patterns, but no path to the requested amount, then employment type is the sole downstream attribute and the requested amount is treated as causally independent of gender.

Once the causal structure is identified, counterfactual traces are generated fitting the SCM to the data in order to learn the structural equations  $F$  under the relationship between variables. In our SCM, the sensitive attribute is always modeled as a root node in  $V$  and is therefore treated as an exogenous variable with respect to the other observed variables. Each structural equation is estimated independently from the training data by fitting a causal mechanism for the corresponding attribute, using its causal parents in the discovered as input. In practice, continuous attributes are modeled with regressors, while categorical attributes are modeled with classifiers.

Therefore, for each trace  $\sigma$ , we build the counterfactual part  $\sigma^{cf}$ . In the loan example, for  $\sigma$  related to a self-employed *female* applicant requesting \$10,000, we generate its counterfactual version  $\sigma^{cf}$  by setting the gender to *male*. Since employment type is causally downstream of gender, the fitted structural equations predict that a male applicant with identical characteristics would be more likely be employed than self-employed. The requested amount remains \$10,000 as it is causally independent of gender. If FCI reveals no causal dependencies between the sensitive attribute and any other case-level attributes, the counterfactual trace  $\sigma^{cf}$  is generated by simply flipping the value of the sensitive attribute.

From each original trace and its counterfactual part, we then extract prefix pairs. If a running prefix  $p$  contains the first  $k$  events of an  $\sigma$ , its counterfactual version  $p^{cf}$  is defined as the prefix containing the first  $k$  events of the  $\sigma^{cf}$ . In this way, each prefix  $p$  is paired with one or more counterfactual prefixes.

## 4.2 Encoding Fairness as a Learning Objective

LTNs ground logical formulas over neural-network outputs and optimize their satisfaction through gradient-based learning. This neuro-symbolic framework use a fuzzy interpretation of first-order logic, where each formula has a truth value in  $[0, 1]$ . The learning objective is defined as the degree to which the model satisfies the logical axioms, i.e., declarative statements that the model must satisfy. Given a set of axioms  $\{\phi_1, \dots, \phi_n\}$ , each axiom receives a satisfaction level  $\phi_i \in [0, 1]$ , measuring how well the current model satisfies it. The satisfaction levels are aggregated into a single scalar value  $\Phi \in [0, 1]$ , and the loss is defined as:

$$L = 1 - \Phi \quad (1)$$

Minimizing  $L$  is equivalent to maximizing the joint satisfaction of all axioms. Since satisfaction levels are computed through differentiable operations, gradients flow back to the model parameters, enabling gradient-based optimization.

We define two logical axioms for the predictor  $f$ : a regression axiom, enforcing accurate KPI prediction for each prefix  $p \in \mathcal{P}$ , and a counterfactual fairness axiom requiring predictions to remain invariant under interventions on the sensitive attribute  $s$ .

To ground both axioms over the space of process prefixes  $\mathcal{P}$ , we define a grounded predicate Eq that measures the degree of similarity between two scalar KPI values  $z$  and  $z'$  using a Gaussian kernel:

$$\text{Eq}(z, z') = e^{-\alpha \sqrt{(z-z')^2 + \epsilon}}$$

The parameter  $\alpha = 0.05$  controls the sensitivity of the similarity measure, and  $\epsilon = 10^{-10}$  is a small constant for numerical stability. This predicate returns a truth value in  $[0, 1]$ , where a value of 1 indicates that the two KPI values are identical and values close to 0 indicate large disagreement. We use a Gaussian kernel as a similarity measure as it maps similarity in  $[0, 1]$ , making it directly interpretable as a fuzzy truth value within the LTN framework.

*Regression axiom.* The regression axiom states that the predictive model  $f$  should accurately predict the target KPI for every process prefix  $p \in \mathcal{P}$ :

$$\forall p \in \mathcal{P} : \text{Eq}(f(p), \mathcal{K}(p)) \quad (2)$$

The satisfaction level  $\phi_{\text{regr}} \in [0, 1]$  of the regression axiom is computed by aggregating the truth values of the Eq predicate evaluated over all prefix-KPI pairs. A value of  $\phi_{\text{regr}} = 1$  indicates that the model predicts the remaining time perfectly for every prefix, while values close to 0 indicate large disagreement between predictions and ground truth. For example, for a running prefix of a loan application, if the model predicts 5.1 days whose true remaining time is 5 days, then  $\text{Eq}(5.1, 5) \approx 0.99$ , which is close to 1. In contrast, if the model predicts 12 days, then  $\text{Eq}(12, 5) \approx 0.49$ , which penalizes the model.

| Dataset    | Sensitive attr.      | Distribution                                                              | Case-level attr.                       | Causal structure                                 |
|------------|----------------------|---------------------------------------------------------------------------|----------------------------------------|--------------------------------------------------|
| bpi12      | gender               | male: 55.83%, female: 44.17%                                              | requested amount                       | gender→requested amount                          |
| hb-gen     | gender               | male: 45.37%, female: 54.63%                                              | age, case type                         | gender→case type                                 |
| hb-age     | age                  | age<65: 89.71%, age≥65: 10.29%                                            | gender, case type                      | age→case type                                    |
| hb-age-gen | age, gender          | age<65: 94.82%, age≥65: 5.18%                                             | age, gender, case type                 | age→case type                                    |
|            |                      | male: 45.99%, female: 54.01%                                              |                                        | gender→case type                                 |
| bpi13i     | organization country | se: 38.34%, pl: 23.35%, us: 10.30%<br>in: 6.50%, be: 6.38%, other: 15.13% | organization involved, impact, product | organization country →product<br>impact →product |

Table 1: Dataset-specific causal setup and sensitive attribute distributions.

*Counterfactual fairness axiom.* The counterfactual fairness axiom requires predictions to remain invariant under interventions on sensitive attributes  $s$ . In process context, cases differing only in protected characteristics, such as gender or age, should receive the same predicted remaining time. Formally, for each prefix  $p \in \mathcal{P}$ , the set  $cf(p, s)$  contains the counterfactual prefixes generated by intervening on  $s$  while keeping all causally unrelated attributes unchanged. The axiom is defined as:

$$\forall p \in \mathcal{P}, p^{cf} \in cf(p, s) : \text{Eq}(f(p), f(p^{cf})) \quad (3)$$

The satisfaction level  $\phi_{\text{fair}} \in [0, 1]$  is computed over all original and counterfactual prefix pairs in the counterfactual event log. Higher  $\phi_{\text{fair}}$  values indicate more consistent predictions across counterfactual pairs, i.e., lower sensitivity to the sensitive attribute. Maximizing the satisfaction of this axiom penalizes prediction differences between an instance and its counterfactual, discouraging reliance on its sensitive attribute. For example, suppose the model predicts 4.8 days for a male applicant and 9.1 days for the counterfactual prefix corresponding to the same case with a female applicant. Then  $\text{Eq}(4.8, 9.1) \approx 0.65$ , indicating poor satisfaction of the fairness axiom.

*Learning objective.* Training combines the two requirements introduced above. For both universal quantification and aggregation of the two axioms, we use the pMeanError aggregator. Given truth values  $v_1, \dots, v_n \in [0, 1]$ , pMeanError is defined as  $1 - (\frac{1}{n} \sum_i (1 - v_i)^p)^{\frac{1}{p}}$ , where  $p = 2$  controls the penalization of low-satisfaction grounding. The overall objective, therefore, aims to find the set of optimal parameters of model  $f$  that maximizes the aggregated satisfaction level  $\Phi$ , obtained by aggregating the values of  $\phi_{\text{reg}}$  and  $\phi_{\text{fair}}$ , assigning equal importance to predictive accuracy and counterfactual fairness. Unlike [9], who adopt a sequential training strategy, we jointly optimize the two axioms, preventing the regression objective from dominating the fairness signal during early training.

## 5 Evaluation

### 5.1 Datasets

We evaluate our approach on three different event logs:

- *BPI Challenge 2012 (bpi12)*: The event log pertains to a Dutch bank loan application process. We focus on the  $A_{-}$  subprocess, enriched with the categorical sensitive attribute *gender*, which is correlated with trace duration.
- *Hospital Billing (hb)*: This event log is sourced from the financial modules of the ERP system of a regional hospital, and contains events related to the billing of medical services provided by the hospital. The traces are enriched with the case-level attributes *age* and *gender*. Three scenarios are considered:

- *hb-gen*: unfair bias in gender, no unfair bias in age
  - *hb-age*: unfair bias in age, no unfair bias in gender
  - *hb-age-gen*: unfair bias in both age and gender
- *BPI Challenge 2013 Incidents (bpi13i)*: An event log of the Volvo IT incident management system. We choose the case-level attribute *organization country* as the sensitive attribute as highlighted in [15].<sup>4</sup>

The *bpi12* and the *hb* event logs have been introduced in [14].<sup>5</sup> The authors preprocessed and adapted the event logs to include sensitive attributes for evaluating group fairness in PPM. The code to reproduce our experiments is available.<sup>6</sup>

Tab. 1 summarizes the dataset-specific causal setup used for counterfactual trace generation. For each scenario, it reports the sensitive-attribute distribution, the case-level attributes considered by FCI, and the causal structure, including the attributes identified as causally downstream of the sensitive attribute.

## 5.2 Experimental Setup

In our experiments, we compare three architectural variants to evaluate the impact of incorporating fairness constraints into a recommendation system:

- **LSTM**: Purely data-driven baseline trained with MSE loss.
- **LTN-Data**: LTN [2] with LSTM backbone and the regression axiom (Eq. 2).
- **Fair-LTN**: LTN [2] where the regression axiom (Eq. 2) and counterfactual fairness axiom (Eq. 3) are jointly optimized with equal weight. Unlike the training strategy adopted in [9], we opt for joint optimization of the two axioms to prevent the regression axiom from dominating the fairness axiom.

The LSTM model is leveraged as a baseline and consists of two stacked layers with a hidden dimension of 128, trained using the Adam optimizer with a learning rate of 0.001 and a batch size of 32. All event logs undergo a preprocessing pipeline implemented in the released code: categorical variables are encoded as integers; numerical attributes are normalized to the unit interval; and missing values are removed. Each prefix is represented as the ordered sequence of its events up to the current point, padded with zeros to the maximum sequence length. Categorical attributes are embedded into 64-dimensional vectors, while numerical attributes are normalized and linearly projected before being passed to the backbone. We train the model for a maximum of 100 epochs with early stopping. While more complex architectures could be considered, we adopt a simplified design to isolate and evaluate the impact of the fairness constraint independently of architectural complexity. The LTN framework is architecture-agnostic, and alternative architectures, such as Transformers or graph neural networks over process traces, could be seamlessly substituted for the LSTM.

<sup>4</sup> Available at [https://data.4tu.nl/articles/dataset/BPI\\_Challenge\\_2013\\_incidents/12693914](https://data.4tu.nl/articles/dataset/BPI_Challenge_2013_incidents/12693914)

<sup>5</sup> Available at <https://zenodo.org/records/15387576>

<sup>6</sup> Available at <https://github.com/FabrizioDeSantis/Fair-PrescriptiveProcessMonitoring>

### 5.3 Train Test Split and Metrics

The event log  $\mathcal{L}$  is temporally split into a training set  $\mathcal{L}^{\text{train}}$  and a test set  $\mathcal{L}^{\text{test}}$ , ordering cases by the timestamp of their first event. The split point  $t_{\text{split}}$  is defined as the earliest time at which 80% of traces in  $\mathcal{L}$  are completed. Traces still running at  $t_{\text{split}}$  form  $\mathcal{L}^{\text{test}}$ , while traces starting after  $t_{\text{split}}$  are excluded to preserve a realistic evaluation setting, where only information available up to  $t_{\text{split}}$  is used. From  $\mathcal{L}^{\text{test}}$ , a running trace log  $\mathcal{L}^{\text{run}}$  is derived by truncating each trace at  $t_{\text{split}}$ , simulating incomplete, ongoing executions against which the quality of recommendations can be assessed. We further extract a validation set composed by the 20% of  $\mathcal{L}^{\text{train}}$ , and use it to tune the hyperparameters, while  $\mathcal{L}^{\text{run}}$  and  $\mathcal{L}^{\text{test}}$  are dedicated to assessing recommendations.

We evaluate predictive performance using the Mean Absolute Error (MAE):

$$\text{MAE} = \frac{1}{|\mathcal{L}^{\text{run}}|} \sum_{p \in \mathcal{L}^{\text{run}}} |f(p) - \mathcal{K}(p)|$$

where  $f(p)$  is the predicted KPI for prefix  $p$ ,  $\mathcal{K}(p)$  is the actual KPI value for  $p$ , and  $|\mathcal{L}^{\text{run}}|$  is the number of prefix in the test log  $\mathcal{L}^{\text{run}}$ . Lower MAE values indicate better predictive performance.

We measure fairness at prediction and recommendation levels. For predictive fairness, we introduce the Counterfactual Mean Squared Error (CF-MSE):

$$\text{CF-MSE} = \frac{1}{|\mathcal{L}^{\text{run}}|} \sum_{p \in \mathcal{L}^{\text{run}}, p^{cf} \in cf(p,s)} |f(p) - f(p^{cf})|^2$$

A lower CF-MSE indicates a fairer model, with a value of zero corresponding to perfect predictive counterfactual fairness.

For prescriptive fairness, we introduce the Counterfactual Recommendation Score (CF-RECC), defined as the fraction of prefixes for which the recommended activity is equal to the one recommended for its counterfactual version:

$$\text{CF-RECC} = \frac{1}{|\mathcal{L}^{\text{run}}|} \sum_{p \in \mathcal{L}^{\text{run}}, p^{cf} \in cf(p,s)} 1[\pi(p) = \pi(p^{cf})]$$

where  $\pi(p)$  denotes the recommended activity for the prefix  $p$  according to the defined policy. A higher CF-RECC indicates a fairer recommender system, with a value of one corresponding to perfect prescriptive counterfactual fairness.

Training uses all prefixes of each trace, whereas testing uses one running prefix per test trace, obtained by randomly choosing  $t_{\text{split}}$  within the trace. Therefore, metrics are averaged over evaluated running prefixes.

### 5.4 Evaluation Through Simulations

Since A/B testing on ongoing processes is infeasible for evaluation purposes, we follow common practice in PrPA and rely on *business process simulation* to assess the effect of following or not following the recommendations under realistic conditions [5]. Each trace  $\sigma'_i$  is a prefix of a corresponding trace  $\sigma_i$  in  $\mathcal{L}^{\text{test}}$ , where  $\sigma_i$  can be seen as  $\sigma'_i \oplus \langle (a'_i, c'_i, t'_i, attr'_{1,i}, \dots, attr'_{m,i}) \rangle \oplus \sigma''_i$ . If the recommendation is the activity  $a$ , we move the simulation to the state reached after performing  $\sigma'_i \oplus \langle (a, c'_i, t_{\text{split}}, attr_1, \dots, attr_m) \rangle$ , namely the recommendation is assumed to occur at the simulation time  $t_{\text{split}}$ . We then simulate the continuation of the

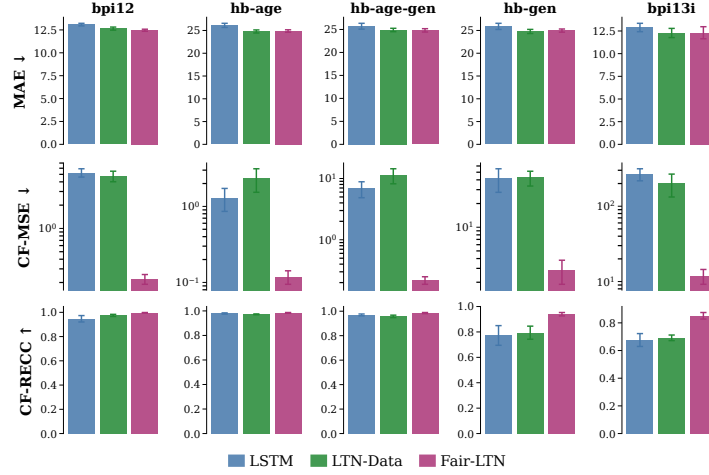


Fig. 2: Performance comparison across models and scenarios. Values represent the mean of 10 independent runs ( $\pm 95\%$  CI). MAE and CF-MSE are error metrics ( $\downarrow$  lower is better) while CF-RECC is a quality metric ( $\uparrow$  higher is better). CF-MSE is reported on a logarithmic scale and axes vary across datasets due to differences in absolute magnitude.

trace to determine how the trace would complete as a result of following the recommendation, comparing then the *trace duration* if a given recommendation has been followed. Since our goal is not only to assess the performance in terms of trace duration prediction, but also the fairness level, we repeat this procedure leveraging the counterfactual version of  $\mathcal{L}^{test}$ , i.e.  $\mathcal{L}_{cf}^{test} = \cup_{\sigma \in \mathcal{L}} \sigma^{cf}$ . This allows us to understand if the recommendations are still effective in a test set in which the value of the sensitive attributes (and what it is correlated with it) changes. As a *baseline* scenario, we assume no recommendation is provided, letting the simulation simulate the timestamps, the attributes and carrying out the process instance. In each setting, each simulation is repeated ten times to mitigate the potential consequences of simulation stochasticity, as the results are reported in terms of mean and 95% confidence interval. Simulations were conducted using the ProSiT simulator [23], using the metrics defined in Sect. 5.3.

## 5.5 Experimental Results

Fig. 2 reports the predictive and fairness performance of all models expressed as means and 95% confidence intervals over 10 independent runs.

*Predictive Performance.* The baselines LSTM and LTN-Data achieve comparable MAE (reported in days) across most configurations, with LTN-Data consistently showing marginal improvements over the vanilla LSTM, suggesting that the LTN regression axiom provides a slightly more effective signal than standard MSE minimization. Fair-LTN achieves similar or improved predictive performance compared to both baselines across all configurations. On *bpi12*, Fair-LTN achieves the lowest MAE (12.46, CI [12.37, 12.55]) compared to 13.08 for LSTM and 12.64 for LTN-Data. On the *hb* datasets, all three models achieve

| Method   | Test Log       | bpi13i           | hb-age            | hb-age-gen        | hb-gen            | bpi12           |
|----------|----------------|------------------|-------------------|-------------------|-------------------|-----------------|
| LSTM     | Original       | 7.23 $\pm$ 0.11  | 107.45 $\pm$ 0.12 | 107.02 $\pm$ 0.14 | 105.39 $\pm$ 0.10 | 5.97 $\pm$ 0.03 |
|          | Counterfactual | 7.38 $\pm$ 0.14  | 106.72 $\pm$ 0.14 | 107.26 $\pm$ 0.12 | 105.63 $\pm$ 0.13 | 6.33 $\pm$ 0.03 |
| LTN-Data | Original       | 7.33 $\pm$ 0.13  | 105.91 $\pm$ 0.10 | 105.97 $\pm$ 0.11 | 105.26 $\pm$ 0.12 | 4.38 $\pm$ 0.02 |
|          | Counterfactual | 7.40 $\pm$ 0.12  | 106.74 $\pm$ 0.13 | 106.08 $\pm$ 0.13 | 107.02 $\pm$ 0.14 | 4.32 $\pm$ 0.08 |
| Fair-LTN | Original       | 7.44 $\pm$ 0.10  | 106.81 $\pm$ 0.13 | 106.51 $\pm$ 0.12 | 105.44 $\pm$ 0.11 | 4.54 $\pm$ 0.06 |
|          | Counterfactual | 7.34 $\pm$ 0.13  | 105.19 $\pm$ 0.11 | 106.78 $\pm$ 0.14 | 106.58 $\pm$ 0.13 | 4.87 $\pm$ 0.01 |
| Baseline | Original       | 11.04 $\pm$ 0.02 | 110.72 $\pm$ 2.12 | 111.28 $\pm$ 1.3  | 112.32 $\pm$ 2.4  | 9.86 $\pm$ 0.02 |

Table 2: Average simulated trace duration (days) of across 10 simulations. The rows labeled “Original” and “Counterfactual” report simulations on the original and counterfactual running prefixes, respectively.

similar MAE values. This indicates that joint optimization preserve accuracy and may act as a regularizer by limiting reliance on spurious correlations between sensitive attributes and the target KPI.

*Predictive Fairness.* The results show a clear and consistent ranking across all configurations. The vanilla LSTM and LTN-Data models exhibit the highest CF-MSE values, indicating that the models trained without explicit fairness constraints learn to exploit sensitive attributes in their predictions. Fair-LTN substantially reduces CF-MSE compared to both baselines across all configurations. The most challenging dataset is *hb-gen*, where LSTM and LTN-Data exhibit high CF-MSE values of 41.53 and 42.20, respectively, indicating a strong influence of the gender attribute on predictions. Fair-LTN substantially reduces this to 2.85 (CI [2.13, 3.70]). The largest absolute improvement is observed on *bpi13i*, where CF-MSE drops from 265.76 (LSTM) and 199.71 (LTN-Data) to 11.83 (CI [9.78, 14.19]) for Fair-LTN. This demonstrates that the joint optimization of the two axioms is effective at enforcing predictive counterfactual fairness. Moreover, the confidence intervals of Fair-LTN are consistently narrower than those of the baselines, indicating that fairness improvements are stable.

*Prescriptive Fairness.* The relationship between predictive and prescriptive fairness reveals an important nuance. On *bpi12*, *hb-age*, and *hb-age-gen*, the LSTM and LTN-Data models already achieve relatively high CF-RECC scores (above 0.95) despite their substantially higher CF-MSE values. This follows from two complementary factors. First, a model can be consistently biased yet still yield consistent recommendations if the bias affects all candidate actions equally without altering their relative ranking. Second, when few candidate activities are proposed for a given prefix, recommendations are forced to be consistent regardless of fairness, trivially inflating CF-RECC scores. In contrast, on *hb-gen* and *bpi13i*, the sensitive attribute has a stronger effect. On *hb-gen*, LSTM and LTN-Data achieve CF-RECC scores of 0.77 and 0.79, respectively, indicating that predictive unfairness translates into prescriptive unfairness when the sensitive attribute affects the relative ranking of candidate actions sufficiently to change the recommended activity. On this dataset, Fair-LTN recovers CF-RECC to 0.94 (CI [0.93, 0.95]). Similarly, on *bpi13i*, LSTM and LTN-Data achieve CF-RECC scores of 0.68 and 0.69, respectively, while Fair-LTN improves this to 0.85 (CI [0.83, 0.87]). Fair-LTN achieves the highest CF-RECC scores, confirming that predictive counterfactual fairness effectively propagates to the prescriptive level.

*Prescriptive Performance.* Tab. 2 reports the average simulated trace duration (in days) for each method, computed as described in Sect. 5.4. For each model,

the *Original* rows refer to simulations on  $\mathcal{L}^{test}$ , while the *Counterfactual* rows refer to simulations on  $\mathcal{L}_{cf}^{test}$ , the *Baseline* corresponds to simulations in which no recommendation is followed. All proposed methods reduce the average simulated duration relative to the baseline, although the magnitude of the reduction depends on the dataset. On bpi12 and bpi13i, the reductions reach 40-56% on bpi12 and 33-35% on bpi13i, while on the three hospital-billing variants the reductions are smaller in relative terms (3% to 7%). Crucially, Fair-LTN obtains nearly the same values with respect to the unfair LTN-Data and LSTM, with absolute differences within  $\pm 0.3$  days on bpi12 and bpi13i and within  $\pm 1.7$  days on the hb variants. In fact, on the Counterfactual rows of hb-age and bpi13i, Fair-LTN even outperforms both baselines. Taken together with the fairness improvements shown in Fig. 2, these findings demonstrate that the proposed framework delivers fair predictions and prescriptions while preserving predictive quality. The following examples illustrate the difference in how individual recommendations may differ between the Fair-LTN and the LSTM baseline.

*Example 1.* For a running process instance of *hb-gen* with prefix [NEW, FIN] and gender set to male, the LSTM baseline recommends RELEASE, but after the counterfactual intervention changing gender to female, it recommends REOPEN. This means that the same billing case would proceed toward release in the original instance, but would be sent back for further handling in the counterfactual one, potentially delaying closure and requiring additional administrative work. Fair-LTN instead recommends RELEASE in both cases, preserving the billing decision. In one seed, this pattern occurs for 19 running instances in the test set.

*Example 2.* In a *bpi12* case with prefix [A\_SUBMITTED, A\_PARTLYSUBMITTED, A\_PREACCEPTED, A\_ACCEPTED], changing gender from female to male makes LSTM switch from A\_DECLINED to A\_FINALIZED, leading the same loan application to a different terminal outcome, whereas Fair-LTN keeps A\_DECLINED in both cases. In one seed, this pattern occurs for 18 running instances in the test set.

## 6 Discussion

### 6.1 Predictive and Prescriptive Fairness

A central finding is that enforcing counterfactual fairness at the predictive level is empirically sufficient to achieve it at the prescriptive level. Since recommendations are determined by the relative ranking of predicted KPI values, a counterfactually fair predictor produces identical rankings for a process instance and its counterfactual, guaranteeing the same recommended activity. This implication is, however, specific to the recommendation policy adopted here and does not necessarily generalize to others. The recommendation policy itself may introduce additional sources of unfairness that are independent of the predictive model. For instance, if the policy incorporates additional decision criteria beyond the predicted KPI, such as resource allocation, sensitive attributes could influence the recommendation through these additional criteria, even when the predictor is perfectly fair. In such settings, predictive fairness alone would not guarantee prescriptive fairness. This can be addressed by adding an axiom that directly encodes counterfactual fairness at the recommendation level, providing a direct signal on the recommendation policy, and enforcing fairness across the pipeline.

## 6.2 Fairness as Regularization

Fair-LTN achieves comparable or lower MAE than both baselines across all evaluated event logs. This suggests that sensitive attributes can act as spurious shortcuts in KPI prediction, with the model exploiting correlations between sensitive attributes and the target KPI present in the training data. By preventing reliance on these shortcuts, the fairness axiom forces the model toward more robust and generalizable predictive patterns, a phenomenon sometimes referred to as fairness for free [10], where fairness constraints simultaneously improve both fairness and predictive performance when sensitive attributes are correlated with process outcomes due to historical bias.

## 6.3 Limitations

While the proposed approach demonstrates promising results, several limitations should be acknowledged. First, the implication from predictive to prescriptive fairness relies on the assumption that the recommendation policy relies exclusively on the predicted KPI. It also relies on the assumption that the transition system is fair, meaning that the set of possible next activities, as well as the control-flow and event-level attributes, are independent from the sensitive attribute. Violations of these assumptions could break this implication and require additional constraints on the recommendation policy. Future work will relax these assumptions by considering the effects of sensitive attributes on control-flow and event-level attributes. Second, the counterfactual trace generation procedure relies on the discovered causal structure, which may be unreliable when the event log is small or when latent confounders are present. The pipeline is also sensitive to the quality of the learned causal structure: omitted dependencies yield incomplete counterfactuals, while spurious ones cause unnecessary attribute recomputation, in both cases optimizing the fairness constraint against imperfect counterfactuals. Furthermore, the learned graph captures statistically supported dependencies but does not determine which causal pathways are normatively admissible; fairness improvements should therefore be interpreted relative to the learned structure, with robustness analysis and domain-constrained causal discovery left for future work. Third, counterfactual fairness can conflict with group fairness. In process contexts where regulatory requirements need statistical parity across demographic groups, the framework would need to be extended with additional group-level constraints. Finally, we focus on remaining time for prediction and trace duration for prescription as KPIs, and counterfactual generation considers one sensitive attribute at a time, leaving interaction effects between multiple sensitive attributes for future work.

## 7 Conclusion

Prescriptive Process Analytics focuses on recommending activities to improve process performance, but these recommendations can be biased by sensitive attributes such as gender, nationality, or age. Prior work in PPM has largely addressed this through group fairness, which ensures statistical equality across the population but offers no guarantee at the individual level: a person may receive a different recommendation solely due to a change in a sensitive attribute,

even when all other case characteristics remain unchanged. Counterfactual fairness addresses this gap directly. This paper proposed a framework for enforcing counterfactual fairness at the prediction level, ensuring that predictions remain invariant to changes in sensitive attributes. The framework generates causally consistent counterfactual traces via FCI-based causal discovery and intervention. Using LTNs, predictive accuracy and counterfactual fairness are encoded as jointly optimized logical axioms, yielding a predictor that is both accurate and fair. Under the recommendation policy adopted in this work, counterfactual fairness propagates to the prescriptive level, yielding consistent recommendations across counterfactual instances.

As future work, we aim to extend the repertoire of KPIs on which we perform experiments and assess results. Although our framework enforces counterfactual fairness through the predictive model, recommendation policies may still introduce bias by prioritizing certain activities, groups or resources. Therefore, an important future direction is to consider constraints directly over policies and also relax the current assumptions that the transition system, control-flow, and event-level attributes are independent of sensitive attributes. We also plan to strengthen the causal component by assessing robustness to causal misspecification, latent confounding, and alternative causal structures. This includes combining causal discovery with domain constraints and conducting ablation studies on how attribute selection and learned causal graphs affect predictive performance, counterfactual fairness, and recommendation quality.

**Acknowledgments.** F. De Santis is supported by the Italian Ministry of University and Research (MUR) under the National Recovery and Resilience Plan (NRRP), Mission 4, Component 1, Investment 4.1, CUP D91I23000080006, funded by the European Union - NextGenerationEU.

## References

1. van der Aalst, W.M.P.: Process Mining - Data Science in Action, Second Edition. Springer (2016)
2. Badreddine, S., d’Avila Garcez, A.S., Serafini, L., Spranger, M.: Logic tensor networks. *Artif. Intell.* **303**, 103649 (2022)
3. Bozorgi, Z.D., Teinemaa, I., Dumas, M., Rosa, M.L., Polyvyanyy, A.: Prescriptive process monitoring based on causal effect estimation. *Inf. Syst.* **116**, 102198 (2023)
4. Burr, T.: Causation, prediction, and search. *Technometrics* **45**(3), 272–273 (2003)
5. De Moor, J., Weytjens, H., De Smedt, J., De Weerdt, J.: Simbank: from simulation to solution in prescriptive process monitoring. In: Proceedings of the 23rd International Conference on Business Process Management (BPM 2025). pp. 165–182 (2025)
6. Dees, M., de Leoni, M., van der Aalst, W.M.P., Reijers, H.A.: What if process predictions are not followed by good recommendations? In: Proceedings of the Industry Forum at BPM 2019 co-located with 17th International Conference on Business Process Management. vol. 2428, pp. 61–72 (2019)
7. Di Francescomarino, C., Ghidini, C.: Predictive process monitoring. In: Process Mining Handbook, pp. 320–346. LNBIP, Springer International Publishing, Berlin-Heidelberg (2022)

8. Gröger, C., Schwarz, H., Mitschang, B.: Prescriptive analytics for recommendation-based business process optimization. In: *Proceedings of the International Conference on Business Information Systems*. pp. 25–37. Springer (2014)
9. Heilmann, X., Manganini, C., Cerrato, M., Belle, V.: A neurosymbolic approach to counterfactual fairness. In: *Proceedings of The 19th International Conference on Neurosymbolic Learning and Reasoning*. vol. 284, pp. 1004–1025. PMLR (2025)
10. Islam, R., Pan, S., Foulds, J.R.: Can we obtain fairness for free? In: *AIES '21: AAAI/ACM Conference on AI, Ethics, and Society, Virtual Event, USA, May 19–21, 2021*. pp. 586–596. ACM (2021)
11. Kubrak, K., Botchorishvili, L., Milani, F., Nolte, A., Dumas, M.: Explanatory capabilities of large language models in prescriptive process monitoring. In: *Proceedings of the 22nd International Conference on Business Process Management (BPM 2024)*. pp. 403–420. Springer Nature Switzerland, Cham (2024)
12. Kubrak, K., Milani, F., Nolte, A., Dumas, M.: Prescriptive process monitoring: Quo vadis? *PeerJ Comput. Sci.* **8**, e1097 (2022)
13. Kusner, M.J., Loftus, J.R., Russell, C., Silva, R.: Counterfactual fairness. In: *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4–9, 2017, Long Beach, CA, USA*. pp. 4066–4076 (2017)
14. Käppel, M., Neuberger, J., Möhrlein, F., Weinzierl, S., Matzner, M., Jablonski, S.: A human-in-the-loop approach for improving fairness in predictive business process monitoring. In: *Business Process Management - 23rd International Conference, BPM*. vol. 16044, pp. 343–360. Springer (2025)
15. de Leoni, M., Padella, A.: Achieving fairness in predictive process analytics via adversarial learning. In: *30th International Conference, CoopIS 2024, Porto, Portugal, Proceedings*. vol. 15506, pp. 346–354. Springer (2024)
16. Mannhardt, F.: *Responsible Process Mining*. pp. 373–401. Springer International Publishing, Cham (2022)
17. Meneghello, F., Middelhuis, J., Genga, L., Bukhsh, Z., Ronzani, M., Francescomarino, C.D., Ghidini, C., Dijkman, R.M.: Optimizing resource allocation policies in real-world business processes using hybrid process simulation and deep reinforcement learning. In: *Proceedings of the 22nd International Conference on Business Process Management (BPM 2024)*. vol. 14940, pp. 167–184. Springer (2024)
18. Padella, A., Mannhardt, F., Vinci, F., de Leoni, M., Vanderfeesten, I.: Experience-based resource allocation for remaining time optimization. In: *Proceedings of the 22nd International Conference on Business Process Management (BPM 2024)*. pp. 345–362. Springer Nature Switzerland, Cham (2024)
19. Pohl, T., Qafari, M.S., van der Aalst, W.M.P.: Discrimination-aware process mining: A discussion. In: *Process Mining Workshops*. Springer, Cham (2023)
20. Shoush, M., Dumas, M.: White box specification of intervention policies for prescriptive process monitoring. *Data and Knowledge Engineering* **155**, 102379 (2025)
21. da Silva, M.C., Fantinato, M., Peres, S.M.: Towards fairness-aware predictive process monitoring: Evaluating bias mitigation techniques. In: *Cooperative Information Systems - 30th International Conference, CoopIS 2024, Porto, Portugal, November 19–21, 2024, Proceedings*. pp. 150–166. Springer (2024)
22. Teinemaa, I., Tax, N., de Leoni, M., Dumas, M., Maggi, F.M.: Alarm-based prescriptive process monitoring. In: *Proceedings of the 16th International Conference on Business Process Management (BPM 2018)*. pp. 91–107. Springer (2018)
23. Vinci, F., Park, G., van der Aalst, W., de Leoni, M.: Prosit: A tool for interactive and transparent process simulations. In: *Proceedings of the Workshops and Satellite Events of the 23rd International Conference on Service-Oriented Computing (ICSOC 2025)*. Springer (2025)