UNIVERSIDAD TECNICA
FEDERICO SANTA MARIA

Causal Process Mining for Temporal Deviations in Event Flows

Submitted in partial satisfaction of the
requirements for the degree of

Doctor en Ingeniería Aplicada

Autor: Fernando Montoya Cáceres
Director de Tesis: J. Ricardo Ñanculef A.
Co-Director de Tesis: Hernán Astudillo R.
Correferente interno: Mauricio Araya L.
Correferente externo: Jorge Munoz-Gama

SANTIAGO DE CHILE
2026

CONSTANCIA DE VALIDACIÓN Y CONFIDENCIALIDAD DE MONOGRAFÍA A REPOSITORIO ACADÉMICO

1.- IDENTIFICACIÓN DEL TRABAJO ACADÉMICO

Tipo de monografía (marcar una opción): ☐ Memoria o trabajo de título ☒ Tesis de Postgrado

Título del trabajo: Causal Process Mining for Temporal Deviations in Event Flows

Nombre del candidato(a): Fernando Antonio Montoya Cáceres

Carrera / Grado: Doctorado en Ingeniería Aplicada

Campus: Casa Central **Departamento:** Dirección de Postgrado

2.- VALIDACIÓN DEL PROFESOR GUÍA/DIRECTOR DE TESIS

Yo, Juan Ricardo Ñanculef Alegria, en mi calidad de profesor(a) guía/director(a) del trabajo académico mencionado anteriormente **DEJO CONSTANCIA** que:

- He revisado esta versión del documento y corresponde a la versión final aprobada del trabajo.
- El trabajo cumple con los requisitos académicos y de formato establecidos por la institución.

3.- EVALUACIÓN DE CONFIDENCIALIDAD POR PROPIEDAD INDUSTRIAL (marcar una opción)

☒ El trabajo **NO contiene** información que amerite confidencialidad y puede ser publicado de inmediato en repositorio con acceso abierto.

☐ El trabajo **CONTIENE** información con potenciales implicancias de propiedad industrial o intelectual y requiere un periodo de confidencialidad (**embargo**) por **(marcar una opción)**:

☐ 6 meses ☐ 12 meses ☐ 2 años ☐ 3 años ☐ 5 años ☐ 10 años

Fundamentación de la necesidad de confidencialidad (obligatorio si se solicita embargo):

4.- FIRMAS

Profesor(a) guía o director(a) de memoria o tesis:

Fecha: 19 - 03 - 2026 **Firma:**

Estudiante o Candidato(a):

Fecha: 12 - 03 - 2026 **Firma:**

Este formulario debe ser insertado como página 2 de la memoria o tesis, completado y firmado por estudiante y profesor(a) antes de la entrega en portal PRISMA de Biblioteca USM.

A mi madre, Lidia, y a la memoria de mi padre, Fernando, por sembrar en mí la curiosidad y la pasión por aprender; su amor y su empuje sostuvieron cada paso de este camino. A mi hija Ignacia, luz que da sentido a cada esfuerzo y razón de mis días. A mis hermanos: Diana, Gladys, Abraham y José Miguel, por el aliento que nunca faltó.

Este logro también es de ustedes.

Agradecimientos

Quiero expresar mi más sincero agradecimiento a mis profesores guía, **J. Ricardo Ñanculef A.** y **Hernán Astudillo R.**, por quienes siento una profunda admiración tanto en lo profesional como en lo personal. Más allá de su invaluable orientación académica y la confianza que depositaron en mí, han sido verdaderos referentes durante todo este camino doctoral. La rigurosidad de sus observaciones, su inagotable disposición al diálogo y, sobre todo, la riqueza de sus visiones complementarias, fueron los pilares que le dieron forma y solidez a esta investigación. Asimismo, valoro enormemente cómo me impulsaron y abrieron las puertas para integrarme de manera activa a una amplia comunidad científica, tanto dentro como fuera del país.

A la **Universidad Técnica Federico Santa María** y a su programa de doctorado, por brindarme el ecosistema institucional ideal para llevar a cabo esta desafiante etapa.

Un agradecimiento especial a **Maribel Olavarría** y **Jaime Arancibia**, gestores del programa. Su calidez humana, excelente disposición y eficiente labor fueron fundamentales para resolver oportunamente los múltiples desafíos de gestión que acompañaron el desarrollo de esta tesis.

A la profesora **Andrea Delgado**, académica de la **Universidad de la República (UdelaR, Uruguay)**, por creer en mi trabajo y facilitar mi integración a la comunidad de **Process Mining**. Su disposición para crear espacios de intercambio y colaboración ha sido una pieza clave en mi aprendizaje.

A **Francisca Raffo**, por la confianza depositada en mí y por haberme brindado, en su momento, la oportunidad de postular al doctorado con el apoyo de la empresa. Su respaldo fue especialmente significativo, no solo porque hizo posible asumir este desafío, sino también porque representó una señal concreta de apoyo a esta etapa de mi formación.

A **Manuel Labbé**, **Claudio Miranda** y **Gonzalo Molina**, por el tiempo compartido y la buena disposición para orientarme. La perspectiva que sumaron desde el sector industrial fue clave para que este trabajo no perdiera de vista las problemáticas y contextos reales.

A mis amigos y compañeros, **Mauricio Hidalgo**, **Gonzalo Rivera** y **Manuel Goyo**. Gracias por las conversaciones interminables, el apoyo incondicional y por empujarnos mutuamente a seguir adelante. Hoy me doy cuenta de que gran parte del valor de estos años no está solo en la tesis, sino en la certeza de saber que nunca estuve solo en este proceso. Más que compañeros de doctorado, formamos un gran equipo, y su amistad hizo que todo este camino fuera profundamente humano y enriquecedor.

A todos ustedes, mi más profunda gratitud.

Resumen

En la gestión de procesos empresariales modernos, caracterizados por su dinamismo y competitividad, las desviaciones temporales constituyen un desafío crítico para la eficiencia operativa. Si bien las herramientas analíticas actuales permiten el monitoreo descriptivo, estas carecen de la capacidad para identificar, interpretar y explicar las relaciones causales subyacentes a dichas ineficiencias, limitando la toma de decisiones efectivas durante las fases de planificación y ejecución.

Esta investigación supera la brecha existente en el análisis de desviaciones temporales mediante el desarrollo de una metodología que integra la Minería de Procesos con la Inferencia y el Descubrimiento Causal. El enfoque desarrollado no solo detecta las anomalías, sino que dota al sistema de interpretabilidad y explicabilidad sobre sus causas raíz. Esto permite optimizar la dinámica de roles y habilita un monitoreo basado en relaciones causales estructurales, proporcionando una comprensión del proceso superior a la ofrecida por las métricas tradicionales.

La metodología se validó mediante un enfoque dual riguroso: en primera instancia, utilizando datos sintéticos de la comunidad científica para pruebas controladas; y posteriormente, a través de una implementación en campo dentro de un entorno empresarial real. Este procedimiento demostró tanto la solidez teórica del modelo como su aplicabilidad práctica para el análisis causal en la industria.

Como investigación aplicada, este trabajo consolida un enfoque integral que reúne la inteligencia causal y la gestión de procesos para gobernar las desviaciones temporales. La principal contribución es un marco que facilita la toma temprana de decisiones basada en el razonamiento causal, permitiendo evaluar acciones focalizadas y estimar de antemano su impacto en la mejora de los procesos, abarcando todo el ciclo, desde la planificación hasta la ejecución.

Palabras clave: *Minería de Procesos Causal; Desviaciones Temporales; Descubrimiento e Inferencia Causal; Explicabilidad Contrafactual; Minería Organizacional.*

Abstract

In the management of modern business processes, characterized by dynamism and competitiveness, temporal deviations constitute a critical challenge for operational efficiency. Although current analytical tools enable descriptive monitoring, they lack the ability to identify, interpret, and explain the causal relationships underlying such inefficiencies, limiting effective decision-making during the planning and execution phases.

This research bridges the existing gap in the analysis of temporal deviations through the development of a methodology that integrates Process Mining with Causal Inference and Discovery. The proposed approach not only detects anomalies but also provides the system with interpretability and explainability regarding their root causes. This makes it possible to optimize role dynamics and enables monitoring based on structural causal relationships, offering a superior understanding of the process compared to traditional metrics.

The methodology was validated through a rigorous dual approach: first, using synthetic data from the scientific community for controlled testing; and subsequently, through an in-field implementation within a real business environment. This procedure demonstrated both the theoretical soundness of the model and its practical applicability for causal analysis in industry.

As applied research, this work consolidates a comprehensive approach that combines causal intelligence with process management to govern temporal deviations. The main contribution is a framework that facilitates early decision-making grounded in causal reasoning, enabling the evaluation of targeted actions and the ex-ante estimation of their impact on process improvement, covering the full cycle from planning to execution.

Keywords: *Causal Process Mining; Temporal Deviations; Causal Discovery & Inference; Counterfactual Explainability; Organizational Mining.*

Contents

Figures

Tables

Chapter 1
Introduction

1.1 Context and Motivation

Current organizations operate in dynamic and highly competitive environments, where the efficiency of business processes is critical. A frequent challenge is the emergence of temporal deviations during execution, even in carefully planned processes [55, 66]. These deviations reduce efficiency, increase costs, and generate delivery delays, directly impacting the final quality of products and services [51].

Process mining offers techniques to discover, monitor, and improve workflows through the analysis of event logs [53, 57, 58]. These techniques enable contrasting actual execution against reference models, analyzing variants, and detecting anomalies or temporal deviations at both activity and trace levels [57, 58].

However, most of these approaches primarily rely on association measures and performance metrics, frequently omitting the explicit modeling of cause-effect relationships [6]. This limitation hinders a deeper understanding of the mechanisms causing deviations and complicates the design of effective causality-based interventions. This issue persists despite recent advances in causal inference [41, 42], explainability through counterfactual reasoning [20], and the acknowledgment of aggregation biases such as Simpson's paradox [19].

Historically, process managers have relied on metrics derived from event logs, model-based simulations (*e.g., discrete-event* `BPMN` *simulations*), fault analysis, and machine learning techniques to predict times and identify patterns [7–10]. While these tools allow evaluation of alternatives and generation of controlled scenarios, they rarely represent explicit causal relationships [7]. As a result, practical decisions often depend on heuristics and tacit expert knowledge, introducing biases and limiting effectiveness [5].

In recent years, attempts have emerged to integrate process mining with causal discovery and inference methods. Some examples include estimating extrinsic delays between pairs of consecutive activities [59] or employing structural discovery frameworks [7, 14]. However, there is still no unified framework that allows not only the identification of deviations but also explanation of their causes and anticipation of the effects of operational changes in event flows.

Temporal deviations originate from diverse sources, such as queues, resource availability, allocation policies, external waiting times, and dependencies between activities [59]. Distinguishing these factors requires models capable of separating spurious correlations from causal relationships, identifying confounding variables, and establishing valid assumptions for causal identification in observational data [7, 41]. In this context, this thesis proposes analyzing event flows as temporally structured sequences, with durations, waiting times, and observed resources, incorporating domain-consistent constraints to enable valid causal inferences.

The practical motivation is that process analysis should not only detect deviations but also explain their causes and support informed decisions to improve performance. A causal approach makes it possible to evaluate how factors such as resource allocation, available capacity, and coordination between roles influence cycle times and deadline compliance [8,10,64]. Unlike purely descriptive methods, this approach introduces clear explanations based on cause-effect relationships and intervention scenarios [20,42], aiming to advance toward models with enhanced causal interpretability and explainability.

1.2 Problem Statement

In the previous section, it was established that temporal deviations in business processes not only affect operational efficiency but also lack a framework explaining their underlying causes. Process mining identifies anomalies in a primarily descriptive manner but does not offer causal interpretations or mechanisms to anticipate their effects [51, 58]. To address this gap, it is necessary to formalize the problem of temporal deviations by explicitly considering activity execution, the temporal structure of traces, and interactions among roles within an explicitly causal approach [7, 20, 41].

Let A be the set of activities, R the set of roles or resources, $T \subset \mathbb{R}$ the temporal domain, and O the space of operational attributes. The set of events E is defined, where each $e \in E$ has projections $\pi_a(e) \in A$ (activity), $\pi_r(e) \in R$ (role), $\pi_s(e) \in T$ (start time), $\pi_e(e) \in T$ (end time), and $\pi_o(e) \in O$ (*associated attributes*). An event log is $L \subseteq E^*$; each trace $\sigma = \langle e_1, \ldots, e_m \rangle \in L$ describes a process instance and satisfies $\pi_s(e_k) \leq \pi_e(e_k) < \pi_s(e_{k+1})$ [52, 53]. The attributes $\pi_o(e) = (o_1, \ldots, o_p)$ include operational variables (*inputs, materials, quality metrics, etc.*) that influence activity durations and the emergence of deviations [9, 56].

Process temporal behavior is also analyzed through variants. The activity projection of a trace is defined as:

$$\nu(\sigma) := \langle \pi_a(e_1), \ldots, \pi_a(e_m) \rangle \in A^*.$$

Two traces belong to the same variant if they satisfy $\nu(\sigma) = \nu(\sigma')$. For $\nu \in A^*$, its equivalence class is defined as $[\nu] := \{\sigma \in L : \nu(\sigma) = \nu\}$ [52, 55]. This notion allows studying how different execution sequences generate distinct temporal patterns [52].

The problem of temporal deviations is observed in two main dimensions. The first dimension corresponds to variability in activity execution. The service time of an event is defined as:

$$\Delta(e) := \pi_e(e) - \pi_s(e).$$

For an activity $a \in A$, considering the set $X_a := \{\Delta(e) : e \in L,\ \pi_a(e) = a\}$, its moments $\mu_a := \mathbb{E}[X_a]$ and $\sigma_a^2 := \mathrm{Var}(X_a)$ describe stability. A high variance σ_a^2 relative to μ_a reflects irregular behavior, complicating planning and increasing operational uncertainty [56, 59].

The second dimension involves deviations in activity flows. For consecutive events e_k, e_{k+1} in a trace σ, with $a_i = \pi_a(e_k)$ and $a_j = \pi_a(e_{k+1})$, the waiting time is defined as $\Delta t_{ij}(\sigma) = \pi_s(e_{k+1}) - \pi_e(e_k)$. A deviation occurs when:

$$\Delta t_{ij}(\sigma) \notin [\,\mu_{ij} - \lambda\sigma_{ij},\ \mu_{ij} + \lambda\sigma_{ij}\,],$$

where μ_{ij}, σ_{ij} are the historical mean and standard deviation of the transition (a_i, a_j), and $\lambda > 0$ is a tolerance threshold [51, 55]. Analysis can extend beyond consecutive pairs. For a trace $\sigma_i = \langle e_1^i, \ldots, e_{m_i}^i \rangle$ and $1 \leq j < k \leq m_i$, the temporal distance is defined as:

$$\delta_i(j, k) := \pi_s(e_k^i) - \pi_e(e_j^i),$$

whose set, ordered lexicographically, forms the vector:

$$x_i^t \in \mathbb{R}^{\binom{m_i}{2}}, \qquad x_i^t = [\,\delta_i(j, k) \mid 1 \leq j < k \leq m_i\,],$$

which summarizes the complete temporal structure of the trace [53]. Thus, deviations in flow are understood both at the local transition level and through temporal relationships between non-adjacent activities within a variant.

These two dimensions are complemented by the organizational dimension. Each event is associated with a role, and interactions between roles influence process times. For $r \in R$, the workload in a trace is defined as $W_r(\sigma) := |\{e \in \sigma : \pi_r(e) = r\}|$, representing the number of events executed by role r in σ. An uneven distribution of W_r generates bottlenecks that increase transition times; these interactions can be modeled using an organizational graph $G_R = (R, E_R)$, where:

$$E_R := \{(r_i, r_j) : \exists\, \sigma \in L,\ e_k, e_{k+1} \in \sigma \text{ with } \pi_r(e_k) = r_i,\ \pi_r(e_{k+1}) = r_j\},$$

and each edge indicates an immediate transfer of work between roles. The expected delay in these transfers, $\Delta t_{r_i r_j} = \mathbb{E}[\,t^{start}(a_j) - t^{end}(a_i) \mid r(a_i) = r_i,\ r(a_j) = r_j\,]$, quantifies how coordination conditions the propagation of temporal deviations [10, 59].

The challenge is to discover and represent structures that explain the relationship among activity variability, flow deviations, and role dynamics. Current process mining techniques detect anomalies but do not explain why they occur [7, 20]. This study adopts a causal perspective integrating operational variables $o_\ell(e)$ to explain $\Delta(e)$ and $\Delta t_{uv}(\sigma)$, employing causal discovery/estimation frameworks [7,14,41,42]. The resulting framework estimates causal effects and their heterogeneity by variant [ν] [19], guiding interventions. Without such a framework, decisions remain based on correlations and expert judgment [5, 41, 42].

1.3 General Objectives

The general objective of this doctoral thesis is to develop and validate a methodology that combines process mining, causal inference, and causal discovery. Its purpose is to identify and explain the causes of temporal deviations in business processes and to optimize role dynamics.

The proposed methodology should provide interpretability, understood as the identification and quantification of causal relationships, and explainability, understood as the ability to answer "*why*" and "*what-if*" questions. Thus, this research aims to contribute to the emerging field of causal process mining and generate practical guidelines for evidence-based management.

1.3.1 Specific Objectives

1. **Design a Theoretical Model** that integrates process mining, causal inference, and causal discovery. This model should identify and quantify causal relationships between temporal deviations and role interactions in business processes, ensuring that results are interpretable by subject-matter experts.

   Description: This objective will focus on creating a conceptual framework that integrates these techniques to address temporal deviations and role dynamics in business processes.

2. **Implement a Prototype** that applies the theoretical model in a real business environment, evaluating its ability to provide interpretability and explainability for temporal deviations and optimize role dynamics.

   Description: This objective will focus on the development and testing of a functional system that uses the concepts of the theoretical model in a specific case study, collecting data and applying the proposed techniques to validate its effectiveness.

3. **Evaluate the Impact of Tactical and Operational Strategies** based on the understanding of causal relationships on the efficiency and effectiveness of work teams in business processes.

   Description: This objective will aim to measure and analyze how interventions based on the new methodology affect team performance, using metrics of efficiency and effectiveness, and adjusting the model and implementation strategies according to the results obtained.

1.4 Research Questions

Based on the background and the problem statement, this doctoral thesis is guided by the following research questions. These questions focus on three main aspects: characterization of temporal deviations, construction of models with causal interpretability, and design of interventions based on causal explanations.

1. **RQ1.** To what extent do temporal deviations in process variants affect the identification and quantification of causal relationships among activities, resources, and roles in business processes?. **Description:** This question addresses the challenge of linking temporal deviations with the underlying causal structure of the process. The goal is to detect patterns in temporal variations and establish how these influence interactions between activities and roles. It seeks to produce robust estimates to objectively assess the impact of deviations on the causal structure.

2. **RQ2.** To what extent does causal interpretability contribute to the understanding of temporal deviations by providing quantitative representations and structured explanations of process flows?. **Description:** This question aims to develop models integrating explicit causal relationships capable of explaining how temporal deviations affect the duration and efficiency of activities and flows. This will enable the assessment of deviations' impact on process performance and the establishment of causality-based metrics.

3. **RQ3.** How can interventions and improvement strategies aimed at mitigating temporal deviations and optimizing role dynamics in business processes be explained in causal terms?. **Description:** This question explores the use of causal explainability in counterfactual scenarios. The objective is to develop a method to evaluate how different improvement actions can be explained through causal relationships and their impacts on the temporal structure of processes and organizational efficiency. The ultimate goal is to propose practical guidelines, grounded in causal explanations, to support decisions regarding process redesign and management.

1.5 Thesis Structure

The remainder of the thesis is organized as follows. **Chapter 2** establishes the theoretical foundation and reviews the key concepts of Process Mining and Causal Reasoning. **Chapter 3** describes the research methodology and presents a hybrid approach that integrates Evaluative Research with the PM^2 methodology. **Chapter 4** constitutes the core of the thesis and presents the compendium of the seven scientific publications aligned with the research objectives. **Chapter 5** provides an overall discussion of the findings and addresses the research questions. Finally, **Chapter 6** synthesizes the main conclusions, examines the hypothesis, and outlines directions for future work. The appendix, placed at the end of the thesis, includes additional collaborative publications that complement the study.

Chapter 2
Background

2.1 Process Mining

In the last two decades, the field of process mining has established itself as a bridge between process science and data science, providing a formal and quantitative framework to study the real execution of organizational processes. Its central objective is to transform event logs stored in information systems into process models capable of describing, diagnosing, and optimizing operational behavior [24, 25].

2.1.1 Purpose and Scope

Unlike traditional Business Process Management (`BPM`) and Business Intelligence (`BI`) approaches, which rely on aggregated indicators such as cycle times or utilization rates, process mining operates at the level of individual events. This enables the reconstruction of complete activity sequences, the analysis of process variants, and the study of both frequent and exceptional behaviors [21]. The paradigm is structured around three main classes of techniques [26]:

- **Process Discovery**, which seeks to automatically build a process model from an event log without prior information about the process structure.

- **Conformance Checking**, which compares a prescribed model against observed behavior in data to detect deviations or non-compliance.

- **Enhancement**, also known as process improvement, which enriches an existing model with additional information regarding time, resources, or costs.

Among these three categories, process discovery occupies a central place. This technique represents the first step in analysis, providing the initial model that will serve as the basis for conformance and enhancement tasks. However, it also represents the most complex challenge.

Process discovery must balance four fundamental quality dimensions: **fitness** (*the model's ability to reproduce data*), **precision** (*avoiding overfitting*), **generalization** (the ability to accommodate plausible but unobserved cases), and simplicity (*understandable, non-redundant models*) [16, 25]. Consequently, process discovery not only defines the starting point for process analysis but also sets the mathematical and computational challenges that the discipline seeks to address today.

2.1.2 Foundations of Process Discovery

Process discovery algorithms form the methodological core of process mining. They allow the automatic construction of models from event logs, transforming operational data into formal representations capturing process logic. This section presents the main approaches and criteria used to evaluate model quality.

2.1.2.1 Process Discovery Algorithms

The **Directly-Follows Graph** (DFG) is the simplest representation. It summarizes direct precedence relationships through a function: $F : A \times A \rightarrow \mathbb{N}$, where A is the set of observed activities. If $F(a, b) > 0$ and $F(b, a) = 0$, an ordering is inferred; if both values are positive, concurrency or flexibility in ordering is interpreted. The DFG can be built quickly and scalably, but naive transitive closures based on F tend to induce cycles and overgeneralization, so it rarely suffices as a final model. [24]

The **Alpha Miner** was the first algorithm to derive Petri nets from relationships observed in an event log. It relies on footprints distinguishing sequence ($a \rightarrow b$), concurrency ($a \parallel b$), and exclusion ($a\#b$), defined based on counts from F. Alpha Miner represented the first demonstration of automatic discovery viability, although it struggles with noise, duplicated activities, and complex loops. Its value is primarily foundational. [24]

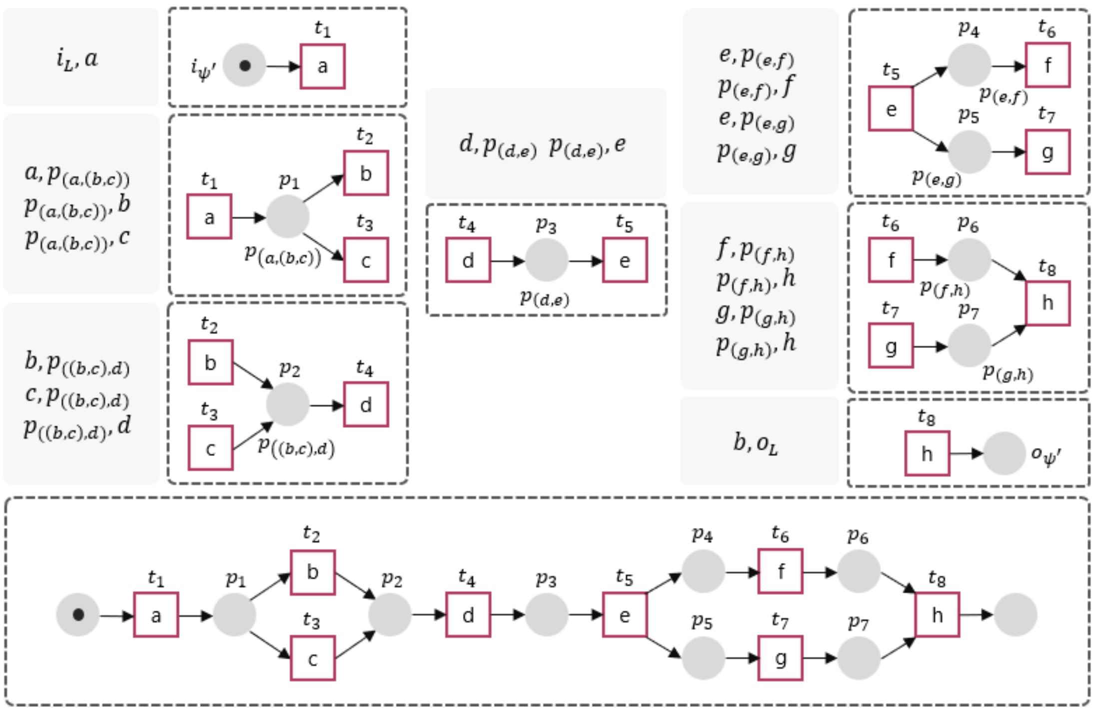

Fig. 2.1: Process model in Petri net notation, obtained by applying the alpha algorithm to an event log of the process $L = [\langle a, b, d, e, f, g, h\rangle,\ \langle a, c, d, e, g, f, h\rangle^4]$.

The **Inductive Miner** represented a decisive advance. It decomposes the log into cuts and constructs process trees using well-defined operators: the sequence operator ($\rightarrow$) establishes a strict ordering between activities, exclusive choice ($\times$) indicates that only one branch executes per instance, concurrency ($\wedge$) allows branches to occur in parallel or any relative order, and the loop operator ($\circlearrowleft$) introduces controlled repetitions of a branch before proceeding. Translating these trees into Petri nets ensures soundness (*formal correctness*) and avoids so-called "spaghetti" models, improving readability and maintainability. [25]

2.1.2.2 Evaluation of Discovered Models

Let L be an event log and M a process model. The quality of M is evaluated across four dimensions: fitness, precision, generalization, and simplicity.

Fitness measures the proportion of traces in L that the model can reproduce [36]:

$$\mathrm{fit}(L, M) = \frac{|\{\sigma \in L \mid \sigma \in \mathcal{L}(M)\}|}{|L|},$$

where $\mathcal{L}(M)$ is the set of traces generated by M.

Precision penalizes excessive behavior allowed by M that does not appear in L. Let $\mathrm{Pref}(L)$ be the set of prefixes in L. For each prefix p, let $C_M(p)$ be possible continuations in M, $C_L(p)$ observed continuations in L, and $w(p)$ a weighting function [38]:

$$\mathrm{prec}(L, M) = 1 - \frac{\sum_{p \in \mathrm{Pref}(L)} w(p)\, |C_M(p) \setminus C_L(p)|}{\sum_{p \in \mathrm{Pref}(L)} w(p)\, |C_M(p)|}.$$

Generalization assesses whether the model abstracts beyond the exact cases recorded in the log. Using a validation subset L'

$$\mathrm{gen}(L, M) = \mathrm{fit}(L', M)$$

Simplicity refers to the structural parsimony of the model. In Petri nets, it can be approximated as $\mathrm{simp}(M) = |P| + |T| + |F|$, where P is the set of places, T the transitions, and F the arcs [36].

2.1.2.3 Integrated Use of Metrics

These four metrics are considered in an integrated manner. The analysis is based on the Pareto front $(\mathrm{fit}, \mathrm{prec}, \mathrm{gen}, \mathrm{simp})$, enabling algorithm comparison and model selection according to objectives such as coverage, restriction, generalization capability, or structural simplicity.

2.1.2.4 Process Variant

A process variant corresponds to a particular trace observed in event log L, considered as a multiset of traces, and let $\sigma = \langle a_1, \ldots, a_n \rangle$ be a trace. We denote by $L(\sigma)$ the frequency of this trace and by $|L|$ the total number of traces. The empirical proportion of the trace is defined as $p(\sigma) = L(\sigma)/|L|$. To summarize the main behavior of the process, we define the coverage of the top k most frequent variants as $C_k = \sum_{\sigma \in \mathrm{Top}_k(L)} p(\sigma)$. Two cases are equivalent if they generate exactly the same trace; each equivalence class represents a variant with multiplicity $L(\sigma)$ [21].

In practice, a few variants typically represent the majority of cases, often displaying a Pareto-like pattern. However, a "variant explosion" is also common. Concurrency generates a combinatorial increase in the number of possible traces, and loops may produce infinitely many distinct executions. Consequently, a log captures only a fraction of the total potential behavior of the process.

Each variant can be characterized through various useful metrics. The length of a trace is defined as $|\sigma|$. The frequency of a specific activity a within a variant can be measured as $n_a(\sigma) = |\{\, i \mid \sigma_i = a \,\}|$. To globally describe the diversity of variants, entropy is used $H(L) = -\sum_\sigma p(\sigma) \log p(\sigma)$. In performance analysis, time measures are often conditioned on specific variants for instance, the average lead time calculated over cases following a variant σ. To manage data complexity, variant-based filtering is typically applied. A simple method is:

$$\mathrm{filter}_{var}(L, \tau) = \Big[\, \sigma \in L \mid L(\sigma) \geq \tau \,\Big],$$

retaining only those variants whose support exceeds a minimum threshold. This filtering improves the clarity and readability of discovered models while preserving the main structure of the process.

The correspondence between a log and a model M is evaluated at the trace level. Let T_M be the set of traces allowed by M. We then define $L_{\mathrm{fit}} = \{\sigma \in L \mid \sigma \in T_M\}$ as the traces fitting the model, and $L_{\mathrm{dev}} = \{\sigma \in L \mid \sigma \notin T_M\}$ as the deviating traces. A simple measure of fitness is the proportion $|L_{\mathrm{fit}}|/|L|$. For a more precise diagnosis of deviations, it is common to use alignments computed for each variant.

Example.

Let A be the alphabet of activities, and L an event log understood as a multiset of traces over A. We write $L = \{\sigma_1^{m_1}, \sigma_2^{m_2}, \ldots, \sigma_k^{m_k}\}$, where each $\sigma_i \in A^*$ is a trace and $m_i \in \mathbb{N}_{>0}$ is its observed multiplicity. The total size of the log is given by $|L| = \sum_{i=1}^{k} m_i$. We define a variant as a unique trace ν (equivalence class defined by sequence equality).

For a trace ν, its equivalence class is $[\nu] = \{\sigma \in L : \sigma = \nu\}$, its absolute frequency is given by $f(\nu) = |[\nu]|$, and its empirical prevalence (relative probability) by $p(\nu) = \frac{f(\nu)}{|L|}$, satisfying $\sum_\nu p(\nu) = 1$. We denote the set of observed variants by $V = \{\nu : f(\nu) > 0\}$. Each variant $\nu = \langle a_1, a_2, \ldots, a_\ell \rangle$ can be associated with relevant structural descriptors.

Specifically:

- **Length:** $|\nu| = \ell$.
- **Direct succession:**

$$E(\nu) = \{(a_i, a_{i+1}) : i = 1, \dots, \ell - 1\} \subseteq A \times A.$$

- **Other possible descriptors** include the set of distinct activities $\mathrm{Act}(\nu) = \{a_i\}$, the multiset of positional jumps, loop patterns, or temporal/resource annotations if traces are enriched.

Consider:

$$L = \{\langle a_1, a_2, a_4, a_5\rangle^3,\ \langle a_1, a_3, a_4, a_5\rangle^2,\ \langle a_1, a_2, a_3, a_4, a_5\rangle^1\}.$$

Then $|L| = 3 + 2 + 1 = 6$ and the observed variants are:

$$\nu_1 = \langle a_1, a_2, a_4, a_5\rangle, \quad \nu_2 = \langle a_1, a_3, a_4, a_5\rangle, \quad \nu_3 = \langle a_1, a_2, a_3, a_4, a_5\rangle,$$

with frequencies $f(\nu_1) = 3,\ f(\nu_2) = 2,\ f(\nu_3) = 1$, and prevalences:

$$p(\nu_1) = \tfrac{3}{6} = 0.5, \quad p(\nu_2) = \tfrac{2}{6} = \tfrac{1}{3} \approx 0.3333, \quad p(\nu_3) = \tfrac{1}{6} \approx 0.1667.$$

Relevant structural descriptors in this example include $|\nu_1| = 4$ with direct succession set $E(\nu_1) = \{(a_1, a_2), (a_2, a_4), (a_4, a_5)\}$, and $|\nu_3| = 5$ with direct succession set $E(\nu_3) = \{(a_1, a_2), (a_2, a_3), (a_3, a_4), (a_4, a_5)\}$. The empirical distribution $\{p(\nu)\}$ allows computing heterogeneity measures, such as the Shannon entropy:

$$H = -\sum_{\nu \in V} p(\nu) \log_2 p(\nu).$$

In the provided example, we calculate step-by-step:

$$-\,0.5 \log_2(0.5) = 0.5 \cdot 1 = 0.5,$$
$$-\,\tfrac{1}{3} \log_2\!\left(\tfrac{1}{3}\right) = \tfrac{1}{3} \log_2 3 \approx \tfrac{1}{3} \cdot 1.5849625 \approx 0.5283208,$$
$$-\,\tfrac{1}{6} \log_2\!\left(\tfrac{1}{6}\right) = \tfrac{1}{6} \log_2 6 \approx \tfrac{1}{6} \cdot 2.5849625 \approx 0.4308271,$$

thus yielding $H \approx 0.5 + 0.5283208 + 0.4308271 \approx 1.4591479$ bits.

Finally, this formulation distinguishes clearly between the statistical dimension (*frequencies and empirical probabilities*) and the structural dimension (*length, successions, patterns*). This separation facilitates formal and applied tasks such as variant clustering, deviation detection relative to frequent variants, probabilistic modeling of traces, and extensions incorporating timestamps or resource attributes (*in which case the definition of a variant extends to enriched traces, e.g.,* (a, t, r) *tuples, and the descriptors must be indexed accordingly*).

2.2 Causal Reasoning

Layer	**Typical Activity**	**Typical Question**	**Example**	**Machine Learning Tasks**
L_1 Associational $P(y\|x)$	Seeing	What is? How would seeing X change my belief in Y?	What does a symptom tell us about the disease?	Supervised / Unsupervised Learning
L_2 Interventional $P(y\|do(x), c)$	Doing	What if? What if I do X?	What if I take aspirin, will my headache be cured?	Reinforcement Learning
L_3 Counterfactual $P(y_x\|x', y')$	Imagining	Why? What if I had acted differently?	Was it the aspirin that stopped my headache?	

Table 2.1: The Pearl's Causal Hierarchy. Each row represents one language, associational (L_1), interventional (L_2), counterfactual (L_3). The columns represent the features of each language. [44]

Causal reasoning does not "*add more statistics*"; rather, it changes the way questions are posed. Table 2.1 serves as a conceptual map: it emphasizes that observing (*associations*), intervening (*actions*), and imagining (*counterfactuals*) belong to distinct levels and are not interchangeable. Operationally, causal reasoning follows these steps: (***i***) a structural causal model (`SCM`) is specified, and the graph's testable implications, specifically d-separations, are validated using observational data at level **`L1`**; (***ii***) based on the graph, it is determined whether an interventional effect can be identified without performing experiments (*using criteria such as back-door, front-door, or do-calculus*), or if an experimental design is necessary for level **`L2`**; and (***iii***) level **`L3`** is reserved for individual attributions and explanations, recognizing that these require stronger assumptions and structural coherence. In summary, this table helps avoid demanding from data answers corresponding to other levels, and clearly documents the transition from observing to intervening, and then to imagining.

2.2.1 Causal Discovery

Causal discovery seeks to reconstruct, either partially or completely, the latent causal structure, typically a Directed Acyclic Graph (`DAG`) $G = (V, E)$, from data, usually observational, along with an explicit set of assumptions [14, 22]. Given a vector $X = (X_1, \ldots, X_d)$ and an i.i.d. sample $D = \{X^{(i)}\}_{i=1}^{n}$, the goal is to identify a graph G whose joint distribution factorizes as:

$$p(x) = \prod_{j=1}^{d} p\Big(x_j \mid x_{\mathrm{pa}_G(j)}\Big),$$

and, whenever possible, orient the edges within the equivalence class of G. This probabilistic interpretation is formalized through the Structural Causal Model (SCM) and the principle of independence of mechanisms [41–44]. According to this principle, causal modules (*the conditional distributions* $p(x_j \mid \mathrm{pa}_G(j))$ *and the error terms that generate them*) are autonomous and independent from each other. This independence justifies the search for causal relationships through regularities or invariances of these mechanisms under changes in inputs or environments.

2.2.1.1 Assumptions in Causal Discovery

Causal discovery generally relies on several key assumptions that allow for reconstructing causal structures [14, 41]:

- **Markov Property and Causal Factorization**: The previously introduced Markov factorization is assumed. Equivalently, in its local form, it suffices to require:

$$X_j \perp\!\!\!\perp X_{V\setminus(\{j\}\cup \mathrm{mb}_G(j))} \;\Big|\; X_{\mathrm{mb}_G(j)},$$

  where $\mathrm{mb}_G(j)$ is the **Markov blanket** of node j, comprising its parents, children, and co-parents (spouses) of its children. This local condition is equivalent to the global form (via d-separation) and factorization.

- **Faithfulness**: It is assumed that all and only the observed independencies reflect the d-separations present in the underlying DAG. This assumption underpins independence-based methods, such as PC, SGS, and IC.

- **Acyclicity and Causal Sufficiency**: Typically, the underlying graph is assumed to be a DAG (*without cycles*), and more strongly, that no latent confounders exist. When these assumptions do not hold, it is recommended to use graphs like MAGs or PAGs along with algorithms such as FCI. Marginalizing latent variables generally breaks the DAG representation, motivating the need for these alternative models.

- **i.i.d. Data and Independence Tests**: In practice, faithfulness is assessed via statistical independence tests ($H_0 : X \perp Y$ or $X \perp Y \mid Z$), controlling Type I errors through p-values, as is standard in independence-based causal discovery [14].

- **Functional Assumptions for Additional Identifiability**: Specific assumptions about functional forms, such as equal error variances in linear Gaussian models, non-Gaussianity in LiNGAM, or non-linearity in ANM, enable clearer identification of causal directions, especially in linear non-Gaussian models (e.g., LiNGAM) and additive noise models, as extensively discussed in recent surveys and domain-specific applications [7, 14, 73].

From a conceptual standpoint, the principle of independence of mechanisms asserts that the distributions P_X and $P_{Y|X}$ (*in the causal direction*) do not encode information about each other; typically, this independence does not hold in the anti-causal direction, generating useful asymmetries for causal discovery.

2.2.1.2 DAG and d-Separation

A DAG $G = (V, E)$ is a causal representation that establishes direct cause-and-effect relationships among variables, indicated by arrows ($i \to j$). This graph specifies which dependencies vanish upon conditioning on certain sets of variables (*interpreted via d-separation*), and organizes data generation into local mechanisms of the form $X_j \leftarrow \mathrm{PA}_j$, defined by conditional distributions $p(x_j \mid x_{\mathrm{PA}_j})$. In this representation, directed acyclic graphs encode both causal ordering and conditional independences via the global Markov property and d-separation [14, 22, 41, 42].

Within this framework, the concept of **d-separation** is crucial, a formal criterion for determining conditional independencies in the graph. Consider a path within the graph: an intermediate node is called a collider if it receives arrows from both directions ($\to k \leftarrow$); otherwise, it is a non-collider node. The criterion of d-separation then states that two sets of nodes A and B are d-separated by a set S if every path between A and B is blocked, either because (***i***) it contains a non-collider node included in S, or (***ii***) it contains a collider node that is neither in S nor has any descendants in S. This condition is formally expressed as $A \perp\!\!\!\perp_G B \mid S$.

Under the Markov principle for DAGs (*global, local, and factorization versions*), d-separation implies conditional independence in the distribution. These three versions are equivalent when the distribution admits a density.

1. **Confounder (common cause)**: $X \leftarrow Z \rightarrow Y$. Graphical interpretation: Z is a non-collider; conditioning on it blocks the only path X–Z–Y. Formally, by d-separation, $X \perp\!\!\!\perp_G Y \mid Z$.

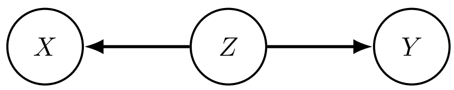

Fig. 2.2: Confounder: Z induces marginal dependence $X \not\perp\!\!\!\perp Y$; conditioning on Z blocks the path.

$$\underbrace{p(x, z, y)}_{\text{local factorization}} = \underbrace{p(z)}_{\text{source}} \underbrace{p(x \mid z)}_{\text{left branch}} \underbrace{p(y \mid z)}_{\text{right branch}},$$

$$p(x, y \mid z) = p(x \mid z)\, p(y \mid z) \;\Rightarrow\; X \perp Y \mid Z.$$

2. **Mediator (chain)**: $X \rightarrow Z \rightarrow Y$. Graphical interpretation: Z is a non-collider; conditioning on it blocks the flow $X \to Y$. By d-separation, $X \perp\!\!\!\perp_G Y \mid Z$.

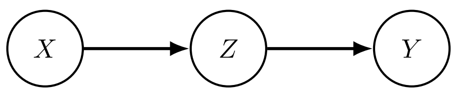

Fig. 2.3: Mediator: the marginal dependence is explained by the path $X \to Z \to Y$; conditioning on Z breaks it.

$$\underbrace{p(x,z,y)}_{\text{local factorization}} = \underbrace{p(x)}_{\text{source}}\ \underbrace{p(z \mid x)}_{\text{transmitter}}\ \underbrace{p(y \mid z)}_{\text{output}},$$

$$p(x,y \mid z) = \frac{p(x)p(z \mid x)p(y \mid z)}{p(z)} = p(x \mid z)\,p(y \mid z) \Rightarrow X \perp Y \mid Z.$$

3. **Collider (v-structure)**: $X \to Z \leftarrow Y$. Graphical interpretation: Z is a collider; *not* conditioning keeps it blocked. By d-separation, $X \perp\!\!\!\perp_G Y$ and $X \not\perp\!\!\!\perp_G Y \mid Z$.

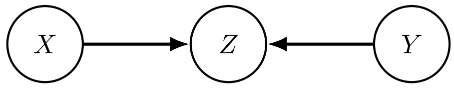

Fig. 2.4: Collider: without conditioning, $X \perp Y$; conditioning on Z (or its descendants) induces dependence ("explaining away").

$$\underbrace{p(x,z,y)}_{\text{local factorization}} = \underbrace{p(x)}_{\text{source 1}}\ \underbrace{p(y)}_{\text{source 2}}\ \underbrace{p(z \mid x,y)}_{\text{convergence}},$$

$$p(x,y) = p(x)p(y) \Rightarrow X \perp Y, \quad p(x,y \mid z) \propto p(x)p(y)p(z \mid x,y) \Rightarrow X \not\perp Y \mid Z.$$

2.2.2 Causal Discovery Algorithms

To derive a Directed Acyclic Graph (`DAG`) from observational data, causal discovery algorithms are employed [7,14,15]. Three main families dominate this field: `PC` (*conditional independencies*), `GES` (*score-based search*), and `LiNGAM` (*functional form*). Each exploits a distinct signal from the joint distribution $P(X)$, delivering either a `CPDAG` (Completed Partially Directed Acyclic Graph, Markov equivalence class), a `PAG` (*Partial Ancestral Graph, accounting for latent/selection variables*), or a fully oriented `DAG`.

2.2.2.1 (PC) Constraint-Based Methods

Under Markov and faithfulness assumptions, the conditional independence (`CI`) pattern determines the Markov equivalence class [14,41]. The basic skeleton rule is:

$$X \text{ and } Y \text{ are adjacent} \iff \nexists\, S \subseteq V \setminus \{X,Y\} \text{ such that } X \perp Y \mid S.$$

The `PC` algorithm does not examine all subsets S; it suffices to consider subsets of neighbors of X or Y, reducing computational cost in sparse graphs [14]. To orient edges, if the skeleton contains $X - Z - Y$ without an $X{-}Y$ edge, and some separator S_{XY} satisfies $Z \notin S_{XY}$, a collider is identified:

$$X \to Z \leftarrow Y.$$

Remaining edges are oriented using Meek's rules (*a complete set of orientation rules preventing cycles and introducing no new independencies*), yielding a CPDAG [14, 41].

In practice, the "oracle" is replaced by independence tests: partial correlation/Fisher's z-test for Gaussian data, G^2/χ^2 tests for discrete data, or HSIC/KCI for nonlinearities. Decisions control a Type I error rate α; Fisher's z-test is classic for Gaussian data and widely used in constraint-based causal discovery [7, 14].

2.2.2.2 (GES) Score-Based Methods

GES maximizes a score balancing fit and complexity. Given data $D = \{x^{(i)}\}_{i=1}^{n}$ and local modules $p(x_j \mid x_{\mathrm{pa}_G(j)}; \theta_j)$, the BIC score is:

$$S(D, G) = \underbrace{\sum_{j=1}^{d} \sum_{i=1}^{n} \log p\Big(x_j^{(i)} \mid x_{\mathrm{pa}_G(j)}^{(i)}; \hat{\theta}_j\Big)}_{\text{log-likelihood (fit)}} - \underbrace{\frac{1}{2} \sum_{j=1}^{d} \dim(\theta_j) \, \log n}_{\text{complexity penalty}}$$

Here, $\dim(\theta_j)$ is the number of parameters in module j (e.g., linear-Gaussian: $|\mathrm{pa}_G(j)| + 1$). Node-wise decomposability arises from causal factorization, allowing local search [14]. GES searches equivalence classes (CPDAG) in two phases: forward (adding edges if $\Delta S > 0$) and backward (removing edges if $\Delta S > 0$). In linear-Gaussian models, score-equivalence holds (*all* DAGs *in the same* CPDAG *score identically*), so optimization naturally occurs in CPDAG-space; BIC is consistent under suitable regularity conditions [7, 14, 41].

2.2.2.3 (LiNGAM) Functional Form-Based Methods

LiNGAM assumes a linear, acyclic model with independent, non-Gaussian noises (*without latent confounders*): $X = B^{\top}X + N, N = (N_1, \ldots, N_d), N_j$ independent, non-Gaussian, $\mathrm{diag}(B) = 0$ [7, 15, 73]. Equivalently: $(I - B^{\top})X = N$, thus $X = A\,N$, with $A := (I - B^{\top})^{-1}$.

Ordering variables causally, B is strictly triangular, and A is triangular with unit diagonal. This structure resolves permutation/scale ambiguities inherent in Independent Component Analysis (ICA) by tying the triangular form directly to a causal ordering.

Identifiability (conceptual): With non-Gaussian independent noises, the linear mixture $X = AN$ is ICA-identifiable up to permutation and scaling. Triangular structure with unit diagonal resolves these ambiguities, yielding:

$$\widehat{B}^{\top} = I - \widehat{W}, \quad \text{where } \widehat{W} \approx (I - B^{\top}) \text{ is ICA-estimated "unmixing".}$$

This mechanism underpins the ability of LiNGAM to recover a fully oriented DAG from observational data under its functional and distributional assumptions [15].

In bivariate cases, asymmetry is clear: if $Y = \alpha X + N_Y$, $N_Y \perp X$, non-Gaussian, no inverse representation $X = \beta Y + N_X$, $N_X \perp Y$ exists (*except Gaussian*). Hence, `LiNGAM` orients $X \to Y$ purely from observational data, a property that has been exploited in applied settings such as manufacturing and business processes [7, 73].

Estimation (two practical paths):

1. **ICA-LiNGAM**: Standardize X. Apply `ICA` for "unmixing" $\widehat{W}$, obtaining (*approx.*) independent $\widehat{N} = \widehat{W}X$. Find permutation P and diagonal scaling D making $D\,\widehat{W}\,P$ unit diagonal and maximally triangular. Define:

$$\widehat{W}^{\star} = D\,\widehat{W}\,P, \quad \hat{B}^{\top} = I - \widehat{W}^{\star}.$$

   Causal ordering follows the triangularity of $\hat{B}$.

2. **DirectLiNGAM**: Uses exogeneity criterion: x_j is exogenous (*no parents*) if independent of all residuals from regressing each x_i onto x_j. For candidate j:

$$r_i^{(j)} = x_i - \hat{\beta}_{ij} x_j, \quad \hat{\beta}_{ij} = \frac{\mathrm{Cov}(x_i, x_j)}{\mathrm{Var}(x_j)},$$

   minimizing a dependence measure $\sum_{i \neq j} I(x_j, r_i^{(j)})$ (e.g., `HSIC`). The selected variable is placed first in causal order; its effect is residualized, repeating until all variables are ordered. Given causal ordering, B is estimated via triangularly restricted `OLS`.

2.2.3 Causal Inference

The causal graph determines who influences whom, whereas causal inference translates this structure into measurable and explainable quantities [42]. The key step involves converting qualitative relationships into interventional distributions, such as $P(y \mid \mathrm{do}(x))$, and representing these via potential outcomes Y_x [41, 45]. Specific estimands, like the `ATE` or `CATE`(z), are identified from the diagram using criteria such as back-door, front-door, or do-calculus, which allow the expression of $P(y \mid \mathrm{do}(x))$ in observational terms [44].

Once identified, these estimands are evaluated through statistical and machine learning techniques. Notably, Double Machine Learning (`DML`) emerges as a contemporary method that leverages flexible predictive models without compromising causal validity [15,49]. Consequently, estimation methods remain tied to the graph's structure, strengthening the connection between theory and practice.

On a deeper level, counterfactuals Y_x enable answering "*what-if*" questions about specific units or subpopulations, thereby completing the cycle among structure, identification, and estimation [20, 43]. The final outcome is a quantitative interpretation of the graph: each arrow corresponds to a causal effect that can be estimated and thus explained.

2.2.3.1 Causal Effect

The notion of causal effect is defined as a contrast between interventional distributions of an outcome variable Y, when an explanatory variable X is fixed at different values. Within the framework of Structural Causal Models (SCM) with Directed Acyclic Graphs (DAG), the **Average Treatment Effect** (ATE) from a baseline value x_0 to a value x_1 is:

$$\mathrm{ATE}_{(x_1,x_0)} = \mathbb{E}[Y \mid \mathrm{do}(X{=}x_1)] - \mathbb{E}[Y \mid \mathrm{do}(X{=}x_0)] = \mathbb{E}[Y_{x_1} - Y_{x_0}]\,,$$

where $\mathrm{do}(\cdot)$ denotes the intervention on X and Y_x is the potential outcome of Y under this intervention [45]. This formulation emphasizes that ATE compares interventional worlds rather than simple observational associations such as $\mathbb{E}[Y \mid X]$. Figure 2.5 (ATE). Two boxplots represent the interventional distributions $P(Y \mid \mathrm{do}(X{=}0))$ and $P(Y \mid \mathrm{do}(X{=}1))$. The shift between central summaries reflects the ATE on an additive scale, while box widths show internal variability.

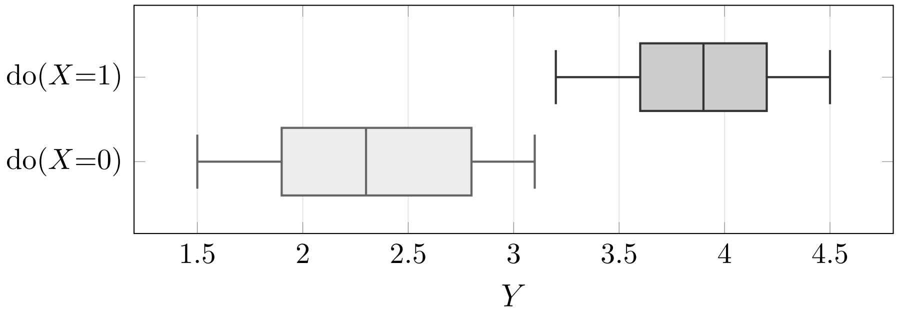

Fig. 2.5: Interventional distributions of Y (*horizontal*). The difference between medians illustrates the average causal effect (ATE).

Global averages may mask context-specific differences. If Z represents relevant covariates (*e.g., age, risk, environment*), the Conditional Average Treatment Effect (CATE) compares the same interventional worlds within each stratum $Z = z$ [20]:

$$\mathrm{CATE}(z) = \mathbb{E}[Y \mid \mathrm{do}(X{=}1), Z{=}z] - \mathbb{E}[Y \mid \mathrm{do}(X{=}0), Z{=}z] = \mathbb{E}[Y_1 - Y_0 \mid Z{=}z]\,,$$

$$\mathrm{ATE} = \mathbb{E}_Z\Big[\mathrm{CATE}(Z)\Big].$$

Figure 2.6 (CATE). Scatter plots in grayscale represent two specific strata: $Z = z_0$ and $Z = z_1$ (*black*). Light points depict samples from $P(Y \mid \mathrm{do}(X{=}0), Z = z)$, and dark points represent samples from $P(Y \mid \mathrm{do}(X{=}1), Z = z)$. Thick bars indicate interventional means, while arrows visually highlight the CATE values within each stratum. When an appropriate set Z satisfying the back-door criterion is available, the ATE can be identified from observational data via:

$$P(y \mid \mathrm{do}(x)) = \sum_z P(y \mid x, z)\, P(z). \tag{2.1}$$

This back-door adjustment connects the causal estimand to purely observational quantities in a way that is robust to measured confounding under standard assumptions [15, 49].

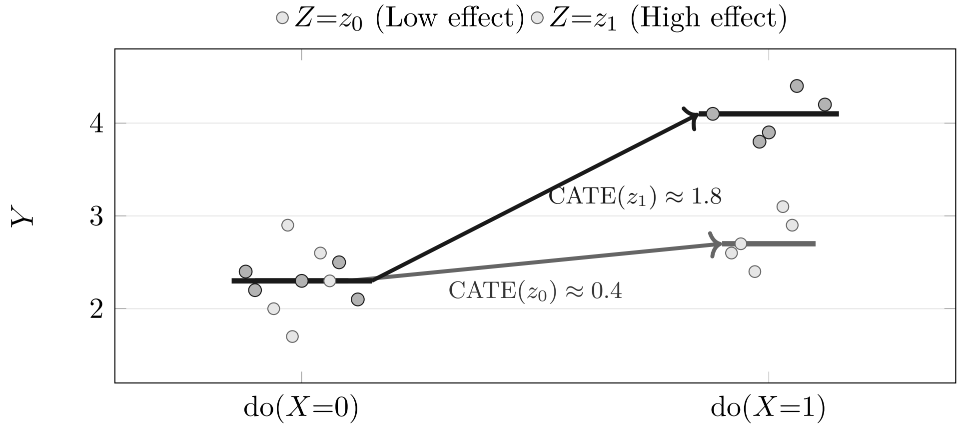

Fig. 2.6: Causal effect heterogeneity across strata of Z. Points represent observations under different interventions, bars indicate interventional means, and arrows denote Conditional Average Treatment Effects (CATE).

2.2.3.2 Double Machine Learning (DML)

After defining causal estimands, the Average Treatment Effect (ATE) and the Conditional Average Treatment Effect (CATE), and establishing their identification via a causal diagram, the subsequent step involves estimation with finite data. Typically, identification is achieved by adjusting for a set of covariates Z that satisfies the back-door criterion (*see* Eq. (2.1)). Identification ensures the causal effect can be expressed as a functional of the observed distribution $Q = f(P(V), G)$ [45].

Estimation entails computing this functional using available data. Here, Double Machine Learning (DML) comes into play (*see* Fig. 2.7). Its goal is to produce robust estimators when employing complex machine learning models [15].

The core idea is to construct an orthogonal score for the parameter of interest. This score is insensitive to first-order estimation errors in the auxiliary components of the identification formula. For the back-door criterion, these components are the conditional outcome regressions $\mu_d(z) = \mathbb{E}[Y \mid X = d, Z = z]$ and the propensity score $e(z) = P(X = 1 \mid Z = z)$. If estimated naively, errors in $\hat{\mu}_d$ or $\hat{e}$ may bias the effect estimator. DML circumvents this bias by combining them into an orthogonal score:

$$\psi(W; \theta, \eta) = \Big(\mu_1(Z) - \mu_0(Z)\Big) + \frac{X}{e(Z)}\Big(Y - \mu_1(Z)\Big) - \frac{1 - X}{1 - e(Z)}\Big(Y - \mu_0(Z)\Big) - \theta,$$

with $W = (Y, X, Z)$ and $\eta = (\mu_0, \mu_1, e)$. This score satisfies the Neyman orthogonality condition $\mathbb{E}[\psi(W; \theta_0, \eta_0)] = 0$, and small perturbations in $\hat{\eta}$ do not affect the primary term. The `DML` estimator is obtained by averaging the score in the sample, with $\hat{\eta}$ computed via cross-fitting. This technique partitions the data into folds, ensuring each prediction is out-of-sample, thus preventing overfitting from contaminating the causal parameter [49].

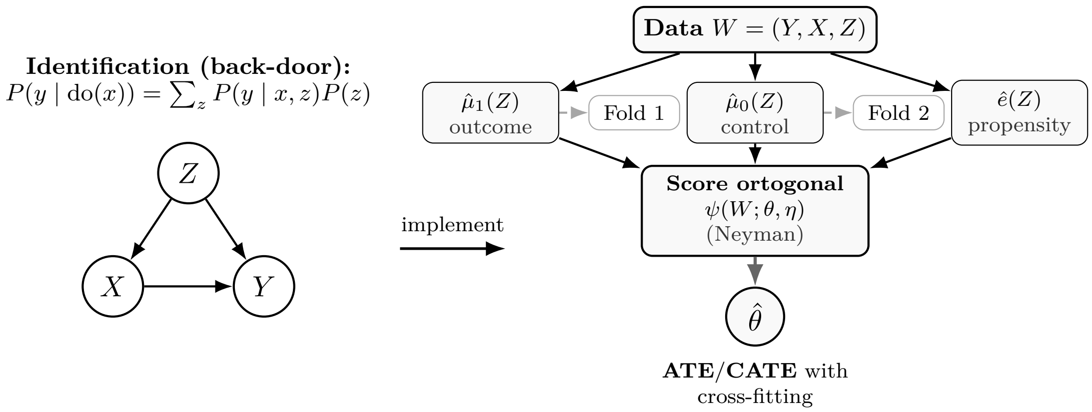

Fig. 2.7: Identification–estimation scheme. **Left:** DAG with Z enables back-door adjustment for causal identification. **Right:** Double ML uses orthogonal score functions and cross-fitting to provide robust estimation of causal effects with ML flexibility.

In the partially linear model:

$$Y = \theta_0 X + g_0(Z) + \varepsilon, \quad X = m_0(Z) + \nu,$$

`DML` appears as the partialling-out procedure. First, Y and X are residualized using predictions $\hat{g}(Z)$ and $\hat{m}(Z)$. Then the residuals are regressed to recover $\hat{\theta}$:

$$\hat{\theta} = \frac{\sum_i \Big(X_i - \hat{m}(Z_i)\Big)\Big(Y_i - \hat{g}(Z_i)\Big)}{\sum_i \Big(X_i - \hat{m}(Z_i)\Big)^2}.$$

Orthogonality ensures that estimation errors in $\hat{g}$ or $\hat{m}$ do not bias the coefficient, preserving the causal interpretation underpinned by the back-door criterion. To capture effect heterogeneity, `DML` extends to `CATE`. A doubly robust pseudo-outcome is constructed:

$$\tilde{Y}^{DR} = \frac{X\,[Y - \hat{\mu}_1(Z)]}{\hat{e}(Z)} - \frac{(1 - X)\,[Y - \hat{\mu}_0(Z)]}{1 - \hat{e}(Z)} + \hat{\mu}_1(Z) - \hat{\mu}_0(Z).$$

Then, $\text{CATE}(z) = \mathbb{E}[\tilde{Y}^{DR} \mid Z = z]$ is estimated. This method follows the logic of the conditional back-door approach, inheriting double robustness: only one of the auxiliary models needs to be correctly specified to maintain validity.

In summary, DML does not identify causal effects; it estimates them. It begins from an already identified expression derived via a DAG and back-door or do-calculus, reformulates it in terms of μ_d and e, and estimates it using advanced machine learning techniques [15]. Through orthogonal scores and cross-fitting, the estimator achieves stability even in high-dimensional settings, thus adhering to the logical sequence of causal analysis: causal diagram → identification → estimation [45].

2.2.3.3 Intervention and Counterfactual

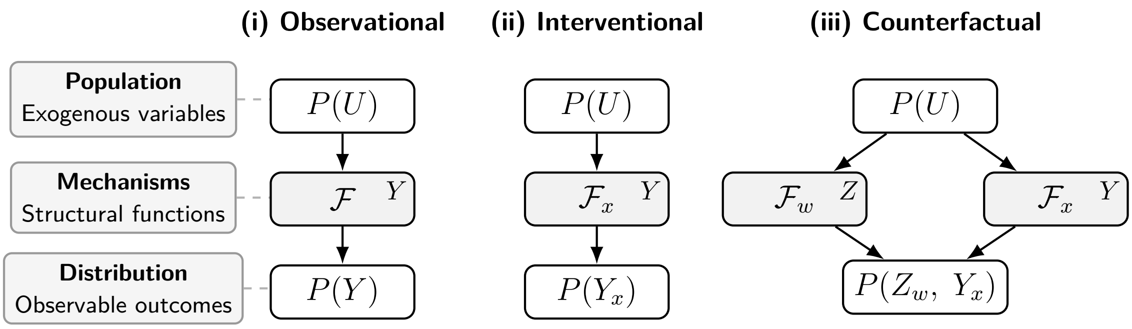

Fig. 2.8: Hierarchy of worlds in an SCM: the same exogenous population $P(U)$ feeds (***i***) natural mechanisms $\mathcal{F}$ inducing $P(Y)$, (***ii***) an intervened mechanism $\mathcal{F}_X$ inducing $P(Y_x)$, and (***iii***) combinations of modified mechanisms $(\mathcal{F}_w, \mathcal{F}_x)$ allowing cross-world queries such as $P(Z_w, Y_x)$.

In causal analysis, distinguishing three types of reasoning, observational, interventional, and counterfactual, is critical (*see* Fig. 2.8) [44]. Each is constructed from the exogenous population $P(U)$, the mechanisms generating the variables, and the resulting distribution.

At the **Observational** level, data arise solely from natural mechanisms $\mathcal{F}$. The outcome is a distribution such as $P(Y)$. This is the realm of traditional statistics and associative methods, as well as the starting point for identification criteria in causal graphs.

The **Interventional** level involves actively modifying the system. By setting a variable $X := x$, its mechanism is replaced by $\mathcal{F}_X$. The resulting distribution is expressed as:

$$P(v \mid \text{do}(x)) = \prod_{V_i \neq X} P(v_i \mid \text{pa}_i)\Big|_{X=x}$$

This operation is equivalent to severing the links that originally determined X. When the graph permits adjustment, this layer can be connected to observational data through simpler expressions like the back-door criterion [42].

The **Counterfactual** level allows one to ask what would have happened under a different action, conditional on what was actually observed. It reuses the same exogenous population $P(U)$ but combines mechanisms across parallel worlds. This permits writing expressions such as $P(Z_w, Y_x)$ or $P(y_x \mid e)$. Its calculation follows the abduction–action–prediction scheme:

$$P(y_x \mid e) \;=\; \sum_u P(y \mid x, u)\, P(u \mid e)$$

First, $P(U)$ is updated with evidence e. Then, intervention $X := x$ is imposed. Finally, the outcome is propagated to yield the result [43]. In summary, $P(y \mid \mathrm{do}(x))$ describes an intervened world, while $P(y_x \mid e)$ answers the "*what if*" question conditioned on factual evidence. The figure illustrates this contrast, showing how each layer articulates population, mechanisms, and distribution.

Chapter 3
Methodology

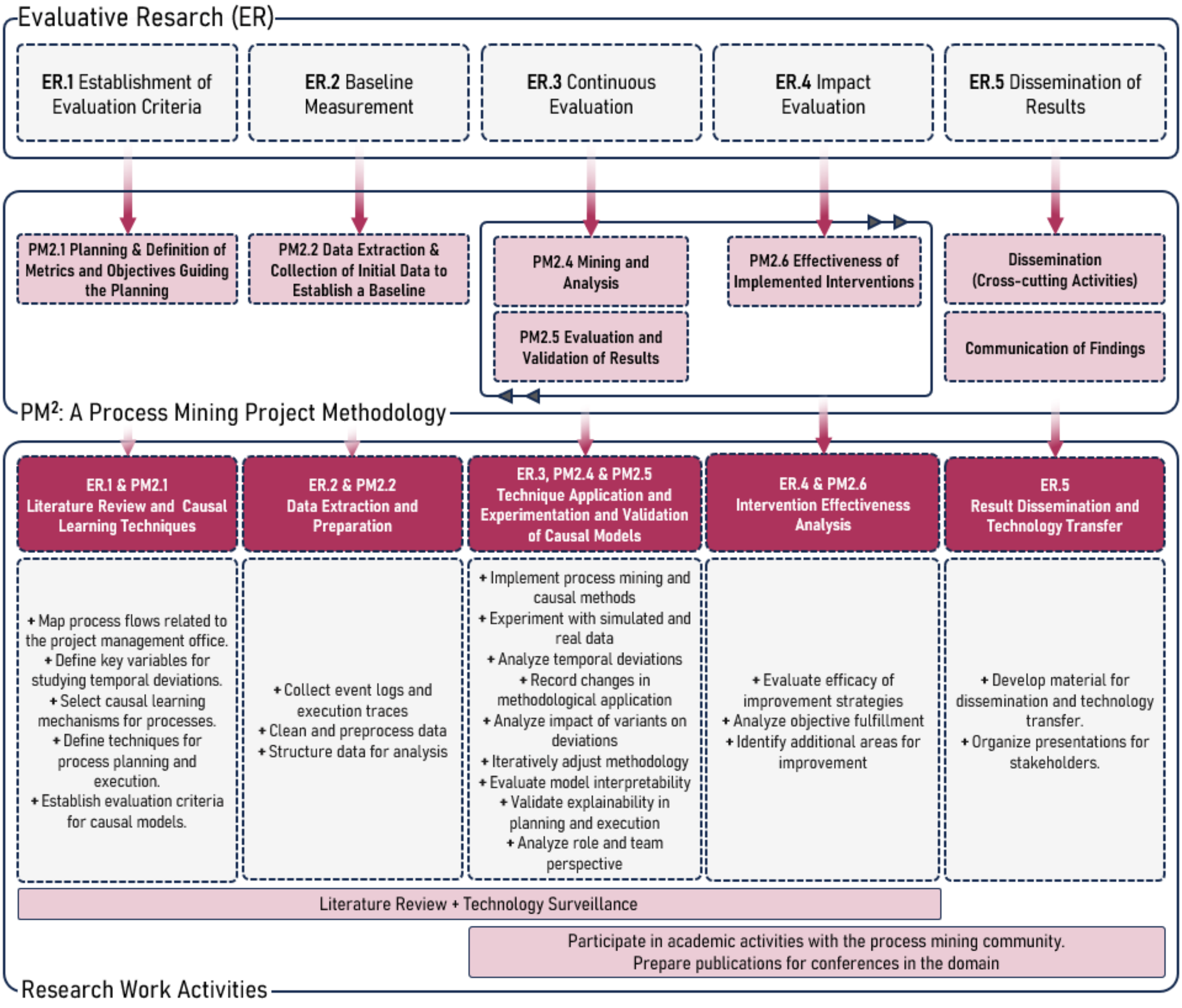

Fig. 3.1: Methodology.

This section presents the methodological framework guiding the compendium. A hybrid approach is adopted, where **Evaluative Research** (`ER`) [27, 28] steers the process through explicit evaluation cycles, while the **PM^2** methodology of process mining provides a clear operational structure: planning, extracting, discovering, analyzing, and validating [29, 30]. The goal is twofold: generating useful evidence for decision-making and producing transferable technology for the organizational context. Figure 3.1 summarizes this interaction.

3.1 Approach and Principles

ER serves as the guiding thread with five essential stages: defining evaluation criteria, establishing a baseline, conducting continuous evaluation, performing an impact evaluation, and finally, dissemination. PM^2 provides concrete activities that operationalize each stage: planning and establishing metrics; extracting and preparing the event log; mining and analyzing the process; validation; and assessing the effectiveness of interventions. This combination ensures each methodological decision is anchored in clear evaluation criteria and generates verifiable artifacts (metric plans, data maps, discovered models, validation reports, and impact reports).

3.2 Integrated ER–PM^2 Flow

1. **Evaluation-oriented planning** (`ER.1  PM2.1`): Objectives, KPIs, and assumptions are defined, setting data scope and quality criteria.

2. **Baseline and data preparation** (`ER.2  PM2.2`): A reference is created from the event log, documenting transformations and inclusion criteria.

3. **Discovery, analysis, and validation** (`ER.3  PM2.4-PM2.5`): The process model is derived, temporal deviations are analyzed, causal techniques are applied, and coherence with domain knowledge and metrics is validated.

4. **Intervention and impact evaluation** (`ER.4  PM2.6`): Controlled changes (rules, sequences, thresholds) are implemented, measuring their effect on KPIs against the baseline, and documenting results and limitations.

5. **Dissemination and transfer** (`ER.5`): Results are consolidated, findings communicated, and materials prepared to facilitate adoption.

Each transition features a decision gate (`D1-D4`): progression occurs only if there is sufficient data quality, metric validity, and evidence of improvement; otherwise, the process returns to the previous step (loop ER.3).

Chapter 4
Compendium of Publications

Overview

Within the framework of this doctoral thesis, the format of a **compendium of publications** has been adopted, which enables coherent integration of scientific contributions derived from research published in international journals and conferences. This format compiles seven articles into a single body, collectively providing empirical and methodological evidence supporting the research objectives.

Each publication addresses a specific dimension of the problem related to temporal deviations in business processes, exploring theoretical and practical aspects of integrating process mining and causal inference. Together, the articles facilitate progress from representing causal relationships in process variants to generating counterfactual explanations, analyzing organizational dynamics, implementing monitoring techniques based on causal graphs, and estimating heterogeneous effects in dynamic contexts.

This compendium reflects a progressive journey from causal interpretability, the capacity to identify and model cause-effect structures in processes, to causal explainability, which provides mechanisms for justifying intervention decisions. This methodological and conceptual advancement directly connects to the three research questions posed in **Chapter 1**, thus fulfilling the thesis's specific objectives.

The following table displays the correspondence among publications, research objectives, and the addressed questions. This mapping clearly illustrates that the articles do not constitute isolated efforts but complementary components of a unified doctoral framework aimed at consolidating **causal process mining** as an approach to managing temporal deviations in event flows.

Publications

Table 4.1: Relationship Between Publications, Objectives, and Research Questions

Paper	Objectives	Research Questions	Contribution
4.1	`SO.1`: Design a theoretical model	`RQ.1`: Identification of causal structures in the presence of deviations	Establishes the **theoretical validity** of causal discovery in process mining. Demonstrates that variant disaggregation is a **statistical prerequisite** to prevent aggregation biases (*e.g., Simpson's paradox*) in the analysis of temporal deviations.
4.2	`SO.2`: Implement a prototype	`RQ.2`: Contribution of causal interpretability to understanding deviations	Validates the practical application of **Causal Inference** for decision-making. Introduces **Counterfactual Explainability** as a mechanism to estimate the impact of "*what-if*" scenarios prior to implementation, thereby reducing managerial uncertainty.
4.3	`SO.1` and `SO.3`: Theoretical model + Impact evaluation	`RQ.1` and `RQ.3`: Causal structure in roles + Interventions	Expands the causal framework to the **Organizational Perspective**. Quantifies, for the first time, how role interaction topologies (*e.g., delegation, collaboration*) act as root causes of temporal inefficiencies, enabling structural team interventions.
4.4	`SO.2`: Prototype applied to monitoring	`RQ.2`: Causal interpretability	Presents the technical architecture of the CaProM prototype. Transforms traditional monitoring (*descriptive*) into **Prescriptive Causal Monitoring**, capable of distinguishing between spurious correlations and true causal anomalies at runtime
4.5	`SO1`, `SO2`, and `SO3`: Theoretical framework, prototype, validation	`RQ.1`, `RQ.2`, and `RQ.3`: Causal structure, interpretability, and explainability	Provides the **comprehensive validation** of the proposed framework. Unifies discovery and inference into a continuous flow and tests its robustness on a massive industrial dataset, quantifying the actual impact of interventions on critical activities.
4.6	`SO.3`: Evaluate impact across diverse scenarios	`RQ.3`: Causal explanations in interventions	Addresses the problem of contextual variability by estimating **Heterogeneous Causal Effects** (**CATE**). Implements Deep Learning (*Multiview Autoencoders*) to segment latent contexts, demonstrating that identical interventions can yield opposing effects depending on the context.
4.7	`SO.1` and `SO.2`: Explanatory model and prototype	`RQ.2` and `RQ.3`: Causal interpretability and explainability	Formalizes the theoretical bridge between **Interpretability** (*seeing the graph*) and **Explainability** (*understanding the why*). Applies the do() operator to generate auditable recommendations, closing the loop between mathematical findings and business actions.

4.1 Causal Graph: Interpretation of Causal Relationships in Temporary Deviations of Business Processes

This first article positions the discussion from aggregated process mining towards a variant-level causal reading ($\mathbf{L}_1 \rightarrow \mathbf{L}_2$). By showing that the causal structure changes with the mixture of variants, it establishes disaggregation as a theoretical condition for any subsequent discovery (**SO.1** , **RQ.1**). In doing so, it reinterprets temporal deviations not only as statistical noise, but as evidence of incompatible causal structures, and formulates this phenomenon as a structural problem that, as far as the reviewed literature extends, has not been explored in process mining.

Publication Summary

Title: *Causal Graph: Interpretation of Causal Relationships in Temporary Deviations of Business Processes*

Authors: Fernando Montoya and Hernán Astudillo

Conference: 49th Latin American Computer Conference (CLEI)

Publisher: IEEE *Xplore*

Year: 2023

Doi: https://doi.org/10.1109/CLEI60451.2023.10345780

Causal Graph: Interpretation of Causal Relationships in Temporary Deviations of Business Processes

Fernando Montoya
Nexus Chile S.A.
&
Universidad Técnica Federico Santa María
Santiago, Chile
fernando.montoya@usm.cl

Hernán Astudillo
ITiSB - Universidad Andrés Bello
Viña del Mar, Chile
&
Universidad Técnica Federico Santa María
Santiago, Chile
hernan@acm.org

***Abstract*—**
Process deviations can be difficult and costly to identify. Therefore, it's imperative for organizations to detect temporary deviations during execution and understand their causal relationships. This enables decision-makers to implement targeted corrective actions. This article presents a method to construct a causal graph for the analysis of process variants, combining techniques from process mining, unsupervised machine learning, and causal discovery and inference. This graph is not susceptible to Simpson's paradox, where aggregating the feature space can lead to incorrect interpretations of causal effects. The technique has been initially validated with a well-known event log, namely a loan application process taken from the BPI Challenge 2017 and containing 16,299 records. This pilot run successfully identified the causal variables and their direction. Wider use of this approach will allow organizations to interpret and estimate the causal effect of an action plan on process variants with temporary deviations.

***Index Terms*—Process mining, Interpretability, Causal graphs, Temporary business process deviations**

I. Introduction

Organizations require tools that provide competitive advantages to their business processes, since industry demands adaptability and differentiation in a dynamic and constantly changing environment. This means that operating units must focus strategic actions on those business processes that support the value chain [1], where the mechanisms that respond to the customer and industry needs are developed. In this scenario, process managers try to execute activity flows in orchestrated and efficient manner, within required times, but operational realities may produce temporary deviations [2] in process execution instances that deviate from their operational guidelines, affecting activities planning, use and availability of operational, technical and financial resources, etc.

The business processes literature includes approaches to identify temporal deviations from several perspectives, but not how to interpret and understand the underlying causal relationships of these deviations. Thus, it is relevant and necessary to have mechanisms that identify and interpret the cause-effect relationships, supporting analysts, managers and process leaders to interpreting the causal dynamics, and allowing the development of focused actions that aim for adaptable and efficient business processes.

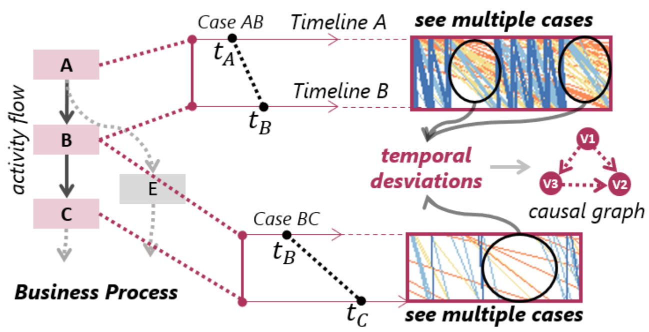

Fig. 1. Characterization of temporal deviations in the business process.

Temporary deviations (TD) of business processes are instances of activities execution that adopt an anomalous temporary behavior that affects the performance of the process execution. Fig. 1 exemplifies business process execution time intervals between activities; e.g. *"$Case_{AB}$"* where A and B are activities and t_B-t_A is the execution time of A; the same happens for *"$Case_{BC}$"*, except that B has a longer execution time t_C-t_B, reflected in the timeline of C. This characterization allows understanding the visualization of the global spectrum of execution times for multiple cases [3], exposing the temporary deviations present in each activity and for several execution instances.

To answer questions related to the causal factors of a time deviations, information is required on the dynamics of the business process execution, in terms of which variables adopt an exogenous role, and therefore may affect variables that adopt an endogenous role. Thus, interpreting the causal direction of the process variables allows knowing in which order to take actions, which in turn requires answering two questions:

- $\boldsymbol{Q_1}$: What causal variables are found in business process execution instances that present temporal deviations?
- $\boldsymbol{Q_2}$: What mechanism would allow to identify and interpret the causal dynamics?

This paper addresses the problem of detecting and interpreting the causal dynamics of temporal deviations in business processes applied in causal graph mechanism, which derives

979-8-3503-1887-6/23/$31.00 ©2023 IEEE

from the domain of Causal Inference [10], the combination of techniques from process mining [4] [5] (that allows to perform process discovery and obtain execution flow variants), and unsupervised machine learning (particularly Gaussian mixtures [13] to identify traces containing temporal deviations).

The remainder of this paper is structured as follows: Section II introduces some preliminary definitions; Section III briefly surveys related work; Section IV explains the proposed technique; Section V presents a case study; and Section VI summarizes and concludes.

II. Some preliminary definitions

This section defines some terms for use along the paper.

Def.1 (Event Log) Let $W = \{a_1, \ldots, a_n\}$ be a set of activities and δ a trace $\in W$, as a sequence of execution of activities $\langle a_1, \ldots, a_n \rangle$; and let $P(\varphi) = \{\emptyset, \{a_1\}, .., \{a_n\}, .., \{a_1, a_2\}, .., \{a_1, a_2, .., a_n\}\}$ be the power set over φ, i.e., the set of possible subsets of the activities of W. An *event log* L is as a multi-set of traces, of the form $L \in P(W^*)$.

Def.2 (Business Process Variant) A tracking variant $\boldsymbol{\sigma}$ is a singular sequence of activities $\langle a_1, \ldots a_n \rangle$ in which all cases $\delta \in L$, corresponding to the same permutation of process activities represented by $\boldsymbol{\sigma}$.

Def.3 (Causality between Variables) A variable X is said to *cause* another variable Y $(X \to Y)$ [11] [12], if adjusting for all confounding factors, an intervention in X results in a change in Y, but an intervention in Y does not necessarily result in a change in X $(X \nleftarrow Y)$. This differs from correlation, which is inherently symmetric, i.e., if X correlates with Y, Y correlates with X, whereas if X causes Y, Y may not cause X.

Def.4 (Causal Graph) A Directed Acyclic Graph (DAG) [13] [14] that represents the causal direction between variables (def.3) as a mechanism for interpreting an existing causal functional relationship. Fig. 2 shows a causal structure model, where each f_i containing a causal factor (*"weight"*) can be an arbitrary function (*"in a fully non-parametric framework"*). Furthermore, ε_i represents the unobserved disturbance (error term), that can exhibit non-Gaussian behavior if the functional relationship between the variables is linear. A causal graph can be obtained by causal discovery algorithms [15].

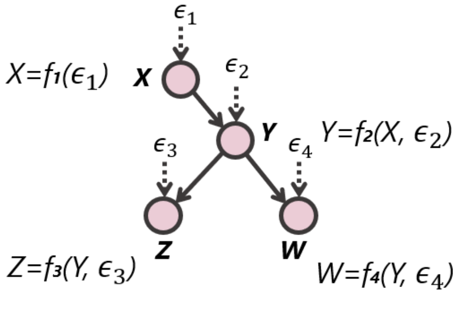

Fig. 2. Causal Graph: Directed acyclic graph

III. Related Work

The identification of temporal deviations in business process TD flows has gained significance in the field of process mining. This emerges as a critical opportunity for the implementation of enhancements. Yet, research has predominantly focused on the task of detecting these deviations, leaving a void in pinpointing the underlying causal relationships that lead to such deviations.

The following addresses two primary domains identified in the literature: one focusing on the identification of temporal deviations and another centered on discerning the causes in business processes.

A. Detecting temporal deviations and selecting process flows as input data

Rogge-Solti et al. [7] propose a Bayesian methodology that originates from a process model expressed in Petri net notation. In this context, the execution flows undergo a filtration process, resulting in the removal of activity relationships that exhibit weak temporal dependencies. This procedure culminates in a condensed business process model, serving as the foundation for identifying deviations through the utilization of a logarithmic probability function.

Zhao et al. [18] and Richter et al. [8] conduct TD detection on a granular selection of pairs of activities that could belong to multiple process variants. In [18], temporal limits are established by the business user, and anomalies are identified when the temporal data falls outside the predefined acceptable bounds. Conversely, in [8], pairs of activities are stored in a hash data structure, and the detection task is carried out using an extended moving average algorithm.

Sontheim et al. [9] and Richter et al. [17] collaborated on research in which z-scoring is employed on the temporal magnitudes of execution within the set of all process variants (def.2), which are represented as a 1xn dimensional vector. In the context of [17], this technique is further expanded by incorporating clustering algorithms such as OPTICS, which facilitates the grouping of temporal deviations (TD).

B. Causal Graph in Business Processes

Qafari et al. [19] address the discovery of an underlying causal structure in a software company. They employ the *Greedy Fast Causal Inference* (GFCI) domain-specific causal discovery algorithm, which conducts a greedy search over a space of potential causal relationships based on the feature environment. This can be further enriched with prior knowledge of causal relationships provided by business experts. The approach makes strong assumptions, such as the independence and identical distribution of the feature space, hence the constructed causal graphs do not presume hidden latent variables (def.4).

Luo et al. [20] present a causal discovery algorithm aimed at tackling flight delay issues within an airline context. They introduce a framework for the automated identification of causal links in flight guarantees based on event sequences. Notably, their approach avoids assuming causal sufficiency, indicating that not all potential shared causes among variables in the model have been observed. This consideration has implications for the arrangement of the causal graph configuration.

Van Houdt et al. [21] similarly propose a technique based on causality theory, employing probabilistic temporal logic to measure cause-effect relationships by considering only the temporal behavior of the activities in the event log (def.1), omitting the feature space that may reveal causal relationships that allow decisions to be made about a set of variables.

IV. Proposed technique - CGTD

In this paper, we address the causal relationships of TD using CGTD (causal graphs of temporal deviations in business process traces) to achieve the feature space illustrated in Fig.3. We direct our methodology towards selecting process variants as per definition (def.2), analyzing each variant independently to circumvent biases inherent to activities shared across different executions.

A. Obtaining a business process execution model

This stage aims to develop a model that reflects an actual execution. For this purpose, specific process discovery techniques are employed within the domain of process mining, as described in [4] (for further details, refer to def.1). Once this model is established, the next step is to identify and extract the various activity flow variants. These variants, taken together, shape and provide a comprehensive perspective of the entire business process model.

B. Focus on process variants

Drawing upon the variations in the process, a disaggregated selection of the feature space is carried out to prevent the generation of causal graphs with potential biases. This paper distinguishes itself from the works cited in III-A, where **TD** events are commonly selected in an aggregated manner, considering common causes for all process variants. Such an approach can lead to incorrect interpretations of the context. This phenomenon, in which aggregated data display outcomes that contradict those of individual groups, is known as the *Simpson's Paradox*, extensively discussed in social and political sciences. It underscores the importance of not overlooking confounding variables and the requirement for thorough analysis.

C. Feature space extraction of temporal deviations

Although the primary aim of this study is not to evaluate the accuracy of **TD** detection techniques, the application of Gaussian models is nonetheless recommended. This stands in contrast to the studies referenced in III-A, which predominantly employ statistical methods (e.g., extensions of moving averages and statistical distances). Gaussian models learn from the feature space $X = \{(x_i, y_i)\}_{i=1}^{n}$ and do not rely on assumptions about **TD**. This obviates the need to reprocess stages IV-A and IV-B for every new set of traces, which can be classified as **TD** by the Gaussian model.

D. Obtaining a causal graph and its interpretability

In this stage, we propose the application of two domain-specific algorithms: LiNGAM [22] and PC [23], to uncover the causal graph in the TD feature space. Fig.4 illustrates the core concept of LiNGAM concerning the linear causal relationship between two variables. Fig.4.*A* displays the regression between X and Y, whereas Fig.4.*B* showcases the dispersion of X concerning the residual ε. It is posited that X causes Y ($X \rightarrow Y$) if X is dependent on ε and Y is not independent of ε. This condition holds if ε exhibits a non-Gaussian behavior, as evidenced by comparing the distributions of Fig.4.*B* and Fig.4.*D*. For causal relationships that do not align with non-Gaussian residual assumptions, it is feasible to apply the PC algorithm [23].

The focus of this paper rests on the significance of structures that provide insight into the causal ordering of variables. This indicates the direction in which decisions should be made in relation to the context. For instance, if X causally affects Y, it implies that, in practice, actions on X will directly impact Y. This notion is reinforced with the presence of causal relationships of the type $X \rightarrow Y \rightarrow Z$. If the situation $X \rightarrow Z \leftarrow Y$ arises, it is understood that Z, within the realm of causality, is deemed a collider variable. In this setting, the business decision-making level is influenced by potential constraints on X and Y. Broadly speaking, it is possible to derive a topology of variables present in the temporal

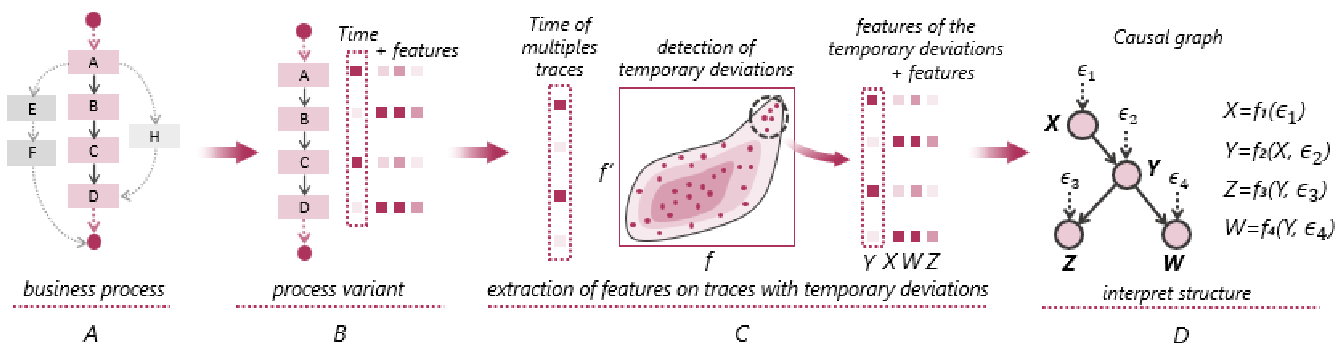

Fig. 3. Stages to identify the characteristics of temporal deviations in business processes from which causal relationships are extracted and interpreted in a causal graph.

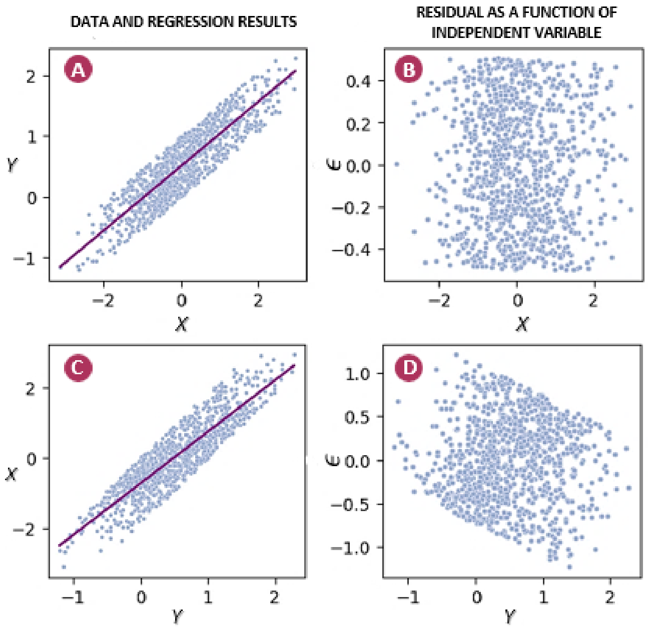

Fig. 4. X is causal in Y, i.e. $X \to Y$, where $Y = f(X) + \epsilon$ and $X \perp\!\!\!\perp \epsilon$

deviations of the process, allowing for the identification of which are endogenous or exogenous within the causal context of the scenario.

V. RESULTS

We apply the steps described in (Fig.3) on an event log that refers to a loan application process (BPI Challenge 2017)[1]. The process variant select in the event log is characterized by the following activities $O\_Create\ Offer$, $O\_Created$, $O\_Sent(mail\ and\ online)$, $O\_Returned$ and $O\_Accepted$, contains 16,299 records. TD detection is applied on the combination of activities as follows if $\{A, B, C\}$ are activities in the process variant, then we will take the combination of time magnitudes between $\{A, B\}$, $\{A, C\}$ and $\{B, C\}$ where we will get vectors for multiple dimension instance 1xn which contains all the temporal relationships of execution between activities.

Extended normalization is applied to z-score, called modified z-score so that by using the median, it exposes the outliers that may be hidden as a result of normalization by the mean, thus enriching the input feature space to the Gaussian mixture model, which has a direct bearing on the broad TD detection. The Gaussian mixture model identifies 521 execution instances that present temporal deviations from the selected variant, which can be seen in (Fig.5)

A. *Causal graphs as a mechanism of interpretability*

In Figure 5 delineates the execution sequences intrinsic to the flow of business processes. These sequences have been pinpointed due to their temporal deviations and are classified under the category "$= 1$". Arbitrarily, the temporal relationship

[1]https://data.4tu.nl/articles/BPI_Challenge_2017/12696884

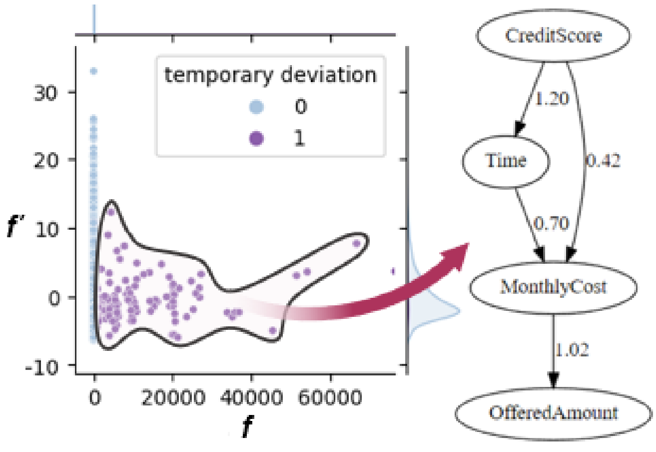

Fig. 5. Causal graph model

between "$(O_Create_Offer, O_Sent\ (mail\ and\ online))$" is chosen. In context with these activities, algorithms within the domain of causal discovery, as referenced in source [22], are employed. The outcome of this process is a causal graph that reveals the underlying causal relationships.

At this point, we can answer the questions I and II:

- $\boldsymbol{Q_1}$: The steps developed, combined with the application of causal discovery techniques, allowed for the identification of which variables constitute the context that shapes the temporal deviations in the execution relationship between activities.
- $\boldsymbol{Q_2}$: [22] and [23] provided a causal adjacency matrix between activities as an output set. Subsequently, this matrix was depicted in a graph retaining the configuration of a *DAG* (see II).

The initial interpretation of the causal graph can be approached from two perspectives:

1) In focusing on the causal factor between the variables, one can discern both the direction and the underlying causal agent. For instance, the variable $Time$ is positively influenced by the variable $CreditScore$. This implies that the dynamics of $CreditScore$ enhance the magnitude of $Time$. In turn, this has an effect on $MonthlyCost$, which subsequently influences $OfferedAmount$. This relationship provides pertinent information regarding the impact that decisions on $CreditScore$ have on the environment.
2) Focus on the topology of the causal network. Identifying which variables play an endogenous or exogenous role is directly related to the complexity of decision-making. For instance,$MonthlyCost$ is a variable exhibiting endogenous behavior in the model, specifically acting as a collider variable influenced by $Time$ and $CreditScore$. This leads to decisions that must balance the impact on $CreditScore$. An added complexity is that $MonthlyCost$ serves as a confounding variable. That is, irrespective of the causal impact actions taken in the $Time \to MonthlyCost$ relationship, the true causal link is $CreditScore \to MonthlyCost$.

VI. Conclusions and Future Work

The causal graphs identified in this study serve as a robust tool for understanding the relationships between temporal deviations in business processes. Furthermore, they provide significant value in terms of strategic knowledge to the organization, thanks to the information recorded daily in operational activities.

The proposed technique allowed for a topology of causal dynamics, which can be obtained with the assistance of multiple causal discovery algorithms. This aids process managers in developing targeted actions, as it is possible to determine the direction of decisions concerning process variables. There are several challenges that still need to be addressed.

This paper has focused on the sequence of activities within the framework of traditional process mining, but it has not delved into causal relationships in processes with an object-centered perspective. This poses a challenge for future research on such processes.

Acknowledgements

This work has been partially supported by ANID under grant PIA/APOYO AFB220004 (CCTVal).

References

[1] Sutarmin, Jatmiko, D.P. Investment Management and Financial Innovations, 13(3), pp. 183–190 (2016)

[2] Florian Richter, Thomas Seidl, Looking into the TESSERACT: Time-drifts in event streams using series of evolving rolling averages of completion times, Information Systems, Volume 84, 2019, Pages 265-282

[3] Denisov V., Belkina E., Fahland D., Van Der Aalst W.M.P. "The performance spectrum miner: Visual analytics for fine-grained performance analysis of processes", CEUR Workshop Proceedings, (2018), pp. 96-100.

[4] Pierluigi Zerbino, Alessandro Stefanini, Davide Aloini,Process Science in Action: A Literature Review on Process Mining in Business Management,Technological Forecasting and Social Change, Volume 172, 2021

[5] Wil Van Der Aalst, Arya Adriansyah, Ana Karla Alves De Medeiros, Franco Arcieri, Thomas Baier, Tobias Blickle, Jagadeesh Chandra Bose, Peter Van Den Brand, Ronald Brandtjen, Joos Buijs, et al. Process mining manifesto. In International Conference on Business Process Management, pages 169–194. Springer, 2011

[6] Alloghani, Mohamed, et al. "A systematic review on supervised and unsupervised machine learning algorithms for data science." Supervised and unsupervised learning for data science (2020)

[7] Rogge-Solti, A., Kasneci, G. Temporal Anomaly Detection in Business Processes. In: Sadiq, S., Soffer, P., Völzer, H. (eds) Business Process Management. BPM 2014. Lecture Notes in Computer Science, vol 8659. Springer, (2014)

[8] Florian Richter, Thomas Seidl, Looking into the TESSERACT: Time-drifts in event streams using series of evolving rolling averages of completion times, Information Systems, Volume 84, pages 265-282, (2019)

[9] Sontheim J., Richter F., Seidl T. Temporal deviations on event sequences, CEUR Workshop Proceedings, 2454, (2019)

[10] Yao, Liuyi, et al. "A survey on causal inference." ACM Transactions on Knowledge Discovery from Data (TKDD) 15.5 (2021): 1-46.

[11] Jonas Peters, Dominik Janzing, and Bernhard Schlkopf. Elements of Causal Inference: Foundations and Learning Algorithms. The MIT Press (2017)

[12] Pearl, Judea. Causality. Cambridge university press, 2009.

[13] Matthew J. Vowels, Necati Cihan Camgoz, and Richard Bowden. D'ya Like DAGs? A Survey on Structure Learning and Causal Discovery. ACM Comput. Surv. 2022

[14] Alessio Zanga, Elif Ozkirimli, Fabio Stella, A Survey on Causal Discovery: Theory and Practice, International Journal of Approximate Reasoning, Volume 151, Pages 101-129, (2022)

[15] Malinsky, Daniel, and David Danks. "Causal discovery algorithms: A practical guide." Philosophy Compass 13.1 (2018).

[16] Reynolds, Douglas A. "Gaussian mixture models." Encyclopedia of biometrics 741.659-663 (2009).

[17] F. Richter, Y. Lu, L. Zellner, J. Sontheim and T. Seidl, "TOAD: Trace Ordering for Anomaly Detection," 2020 2nd International Conference on Process Mining (ICPM), Padua, Italy, 2020, pp. 169-176

[18] D. Zhao, Z. Zhou, Y. Wang and W. Gaaloul, "Detecting Temporal Anomaly and Interestingness in Timed Business Process Models," IEEE International Conference on Web Services (ICWS), Beijing, China, 2020

[19] Qafari, M.S., van der Aalst, W. Root Cause Analysis in Process Mining Using Structural Equation Models. In: Del Río Ortega, A., Leopold, H., Santoro, F.M. (eds) Business Process Management Workshops. BPM 2020. Lecture Notes in Business Information Processing, vol 397. Springer (2020)

[20] Qian Luo, Lin Zhang, Zhiwei Xing, Huan Xia, Zhao-Xin Chen, "Descubrimiento causal del proceso de servicio de vuelo basado en la secuencia de eventos", Journal of Advanced Transportation, 17 páginas, 2021.

[21] Van Houdt, G., Depaire, B., Martin, N. Root Cause Analysis in Process Mining with Probabilistic Temporal Logic. In: Munoz-Gama, J., Lu, X. (eds) Process Mining Workshops. ICPM 2021. Lecture Notes in Business Information Processing, vol 433. Springer, (2022).

[22] Shimizu, Shohei. "LiNGAM: Non-Gaussian methods for estimating causal structures." Behaviormetrika 41, pag. 65-98. (2014)

[23] Spirtes, P., Glymour, C., and Scheines, R. Causation, Prediction, and Search. The MIT Press, Cambride, MA, 2nd edition. (2000)

4.2 Counterfactual Explanability: An Application of Causal Inference in a Financial Sector Delivery Business Process

Building on that structural basis in Paper.1, the second article advances from graph validity towards counterfactual explainability (**L**$_3$). It integrates causal discovery and causal inference to compare decisions as "*what-if*" scenarios, articulating the link between the graph and the business language (**SO.2**, **RQ.2**). In the literature, this contribution broadens the analysis of decisions in processes, traditionally based on predictions or isolated coefficients, by incorporating an assessment grounded in explicit counterfactual alternatives.

Publication Summary

Title: *Counterfactual Explanability: An Application of Causal Inference in a Financial Sector Delivery Business Process*

Authors: Fernando Montoya, Esteban Berríos, Daniel Díaz and Hernán Astudillo

Conference: 42nd IEE International Conference of the Chilean Computer Science Society (SCCC)

Publisher: IEEE *Xplore*

Year: 2023

Doi: https://doi.org/10.1109/SCCC59417.2023.10315742

Counterfactual Explanability: An Application of Causal Inference in a Financial Sector Delivery Business Process

Fernando Montoya
Nexus Payment Systems Spa.
Santiago, Chile
&
Fundación Instituto Profesional Duoc UC
Santiago, Chile
&
U. Técnica Federico Santa María
Santiago, Chile
fernando.montoya@usm.cl

Esteban Berríos
Fundación Instituto Profesional Duoc UC
Santiago, Chile
es.berriosp@duocuc.cl

Daniela Díaz
IT Solution Spa.
Santiago, Chile
ddiazb4@correo.uss.cl

Hernán Astudillo
ITiSB - Universidad Andrés Bello
Viña del Mar, Chile
&
U. Técnica Federico Santa María
Santiago, Chile
hernan@acm.org

***Abstract*—**
Service owners, administrators, and business process analysts are constantly confronted with a dynamic of operational changes aimed at aligning business processes with the demands and requirements of the environment. This compels them to take actions that enable the efficient redirection of available efforts and resources. The challenge lies in obtaining a structure of the variables and causal relationships involved in the actual execution of the process, enabling the evaluation and response to potential operational scenarios while minimizing the uncertainty associated with selecting an improvement plan for a specific business process flow. This paper presents a method and its application for evaluating decision-making in the context of business processes, specifically regarding the topological direction and counterfactual explainability of variables that have a causal effect in potential improvement scenarios. This approach is achieved by combining techniques derived from causal discovery and inference, as well as process mining. The technique has been validated through a real-world case in the Chilean financial industry, specifically in a credit card delivery process. During this study, the underlying causal relationships in the operational flow were successfully identified, enabling process managers and analysts to evaluate the causal effect of interventions (counterfactuals) and select the most efficient and goal-aligned improvement actions. A broader application of this approach allows organizations to justify the estimation of the causal effect of an action plan through counterfactual reasoning.

***Index Terms*—Discovery and causal inference, process mining, explainability in business processes**

I. Introduction

Keeping business processes aligned with the objectives of an organization [1] requires conducting a constant assessment of the operational reality. This entails that process owners and managers must analyze various improvement actions in response to different environmental scenarios. To efficiently evaluate and select improvement actions in the operational flow, it is necessary to possess a solid understanding of the behavior of variables involved in the process. This knowledge enables an understanding of how these variables interrelate and influence the performance and outcomes of the process. The evaluation of the impact of potential decisions before their adoption can be costly and complex. Currently, there are process simulation mechanisms that utilize domain-specific notations, such as **BPMN** [2], which enable the recreation of a controlled reality and observe how operations flow.

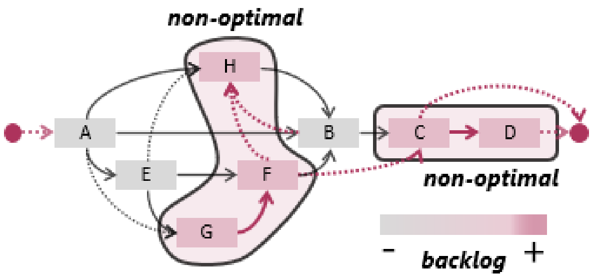

Fig. 1. Characterization of non-optimal activity flow in the business process, the temporary effect is exemplified as a bottleneck.

These mechanisms also offer the possibility to optimize each scenario under analysis [3]; however, it is important to note that many of these simulation mechanisms rely on analyzing correlations between variables. This can lead to a misinterpretation when explaining the underlying causal relationships of the process. It is crucial to involve domain experts to ensure accurate understanding. Correlations can provide useful information about interactions between variables, but they do not always indicate a direct causal relationship.

A common scenario encountered in business processes is illustrated in Fig.1, where two scenarios of irregularities

979-8-3503-1389-5/23/$31.00 ©2023 IEEE

are depicted, which can arise in the variations of activity execution. While both scenarios have a longer execution time compared to other process flows, it is important to note that this does not necessarily imply identical underlying causes. In fact, there may be discrepancies in the causal topology, i.e., the relationships and causal directions among the variables involved in each operational flow. These irregularities can have a significant impact on the overall process flow and may lead to delays, errors, or failure to meet established objectives. Therefore, it is crucial to identify and understand these irregularities in order to address them properly and improve process efficiency and quality.

A. *Counterfactuals and Explainability*

In this context, it is paramount to have mechanisms that facilitate the early interpretation of the analyzed scenario, thereby enabling the explicability of the effects derived from different decision-making processes. A valuable instrument to accomplish this is the utilization of counterfactuals, a fundamental concept within the realm of causal inference [5].

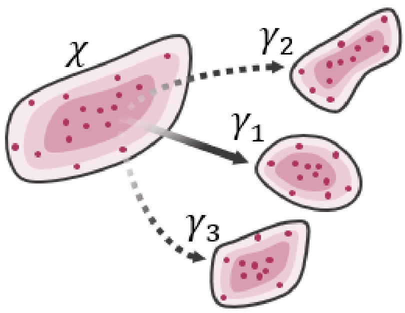

Fig. 2. γ_2 y γ_3 represent distributions. For each possible intervention in χ, γ_1 is the observed effect, given that χ is causal of γ_1 ($\chi \rightarrow \gamma$).

When considering a running business process, questions may arise such as: *"What would have happened if we had made a different decision at a specific point in the process?"*. The concept of counterfactuals allows us to explore that question and understand how the outcome would have been if we had followed an alternative path, In additional terms, *"In what manner can we develop causal explainability of counterfactual scenarios?"*.

Fig.2 depicts the causal relationship between variables χ and γ, denoted as $\chi \rightarrow \gamma$. This indicates that when an intervention is conducted on χ, an effect is observed on γ. Within this framework, γ_1 corresponds to the observed effect (factual scenario), whereas γ_2 and γ_3 represent potential outcomes (counterfactual scenarios) resulting from hypothetical interventions made in χ. It is crucial to note that this causal relationship holds true only if a prior causal connection has been identified between the variables.

The identification of causal relationships among a set of variables can be addressed within the domain of causal discovery [6], a topic that has gained relevance in recent times. Furthermore, it has been explored as a tool for interpreting machine learning models [7].

B. *Challenge and Questions*

The primary objective of this paper pertains to the task of counterfactual explainability. In this context, the utilization of causal graphs assumes a pivotal role as instruments for interpreting the intrinsic causal relationships within operational workflows. To accomplish this objective, methodological frameworks that integrate the domains of causal discovery and inference [5], along with the field of process mining [4], are leveraged. By employing a combined application of these techniques, a comprehensive understanding of causal scenarios in the context of business process flows is achieved.

The application of these techniques is validated by estimating causal effects for a series of decisions in a credit card delivery process in a real-world financial sector environment. The aim is to provide answers to the following questions:

- Q_1 : What are the key variables that influence each scenario under analysis in the credit card delivery process?
- Q_2 : How are these variables related to each other, and what are the underlying causal relationships?
- Q_3 : What causal effects do certain decisions have on the process?

The remainder of this paper is structured as follows: Section II preliminary definitions; Section III surveys related work; Section IV explains the proposed technique; Section V presents a case study; and Section VI summarizes and concludes.

II. Some Preliminary definitions

The following are some key concepts in the domain of causal inference and process mining:

Def.1 (Causal Effect) Causal effect can be defined as the impact of a treatment T (an action or intervention applied to a group of individuals in the study) by comparing the outcomes of different treatment values on a population. A causal measure that quantifies the average impact of a treatment or intervention is **ATE**, denoted as $\boldsymbol{\tau}$ = **ATE** (Average Treatment Effect).

$$\boldsymbol{\tau} = \mathbb{E}\left[Y_i(1) - Y_i(0)\right] \tag{1}$$

$Y_i(1)$ and $Y_i(0)$ represent potential outcomes for each unit $\boldsymbol{i}$, where $Y_i(1)$ is the potential outcome if unit $\boldsymbol{i}$ receives treatment $T = 1$, and $Y_i(0)$ is the potential outcome if unit $\boldsymbol{i}$ does not receive treatment $T = 0$.

Def.2 (do-operator $p(\cdot \mid do(x))$) It is a common probability distribution that adheres to the same rules of conditioning $p(y \mid x)$ and marginalization $p(x)$. The operator $do(x)$ indicates that we are working with an interventional distribution rather than an observational one. We can define expectations $\mathbb{E}$ as causal effects of interventions:

$$\mathbb{E}[Y \mid do(X = x)] := \sum_y y \cdot p(y \mid do(x)) \tag{2}$$

When an intervention is performed on X by setting it to a specific value x, the conditional probability of Y, denoted as $p(Y \mid do(X = x))$, is determined, thereby mitigating any potential external influences or confounding factors.

Def.3 (Causal Model) A causal model has the purpose of interpreting how a variable or a set of variables influence the occurrence or change of another variable of interest. As a mechanism for interpretability, a directed cyclic graph $\boldsymbol{\mathcal{G}}$ can be employed, denoted as follows:

$$P\left(X_1, \ldots, X_n\right)=\prod_{i=1}^n P\left(X_i \mid \mathbf{pa} \boldsymbol{\mathcal{G}}_{\boldsymbol{i}}\right) \tag{3}$$

Where $\mathbf{pa}\boldsymbol{\mathcal{G}}_{\boldsymbol{i}}$ represents the parent nodes (*causal*) of the variable node X_i (*effect*) in the directed acyclic graph $\boldsymbol{\mathcal{G}}$ (DAG).

Def.4 (Counterfactual) Counterfactuals are hypothetical scenarios about a causal relationship and explain the effects of an action. To achieve this, a method is required to interpret the causal relationships, such as $\boldsymbol{\mathcal{G}}$ (DAG). Let X_1, X_2 be an identifiable causal relationship represented as $X_1 \rightarrow X_2$, we can perform a counterfactual of the form $p(X_2 \mid do(X_1 = x))$, where x is a value from the hypothesis set. In general terms, we can denote that when performing a counterfactual of $X_j = x$ in $\boldsymbol{\mathcal{G}}$, we have:

$$\underbrace{P_{(X_j=x)}(\mathbf{X})}_{\text{truncated distribution}}=\prod_{i \neq j}^n P\left(X_i \mid \mathbf{pa} \mathcal{G}_i\right) \times \underbrace{I_{\{x\}}\left(X_j\right)}_{\text{counterfactual}} \tag{4}$$

The distribution generated by the intervention $do(X_j = x)$ is a truncated factorization $P\left(X_1, \ldots, X_n \mid do(X_j = x)\right)$, where the Indicator I_x indicates that $p(X_j) = 1$ if $X_j = x$, otherwise $p(X_j) = 0$.

Def.5 (Trace Variant) A tracking variant $\boldsymbol{\sigma}$ is a singular sequence of activities $\langle a_1, \ldots a_n \rangle$ in which all cases $\boldsymbol{\delta} \in \boldsymbol{L}$, corresponding to the same permutation of process activities represented by $\boldsymbol{\sigma}$.

Def.6 (Event Log) Be a set of activities $W = \{a_i\}_{i=1}^n$ and $\boldsymbol{\sigma}$ a trace $\in W$, as an execution of activities $\langle a_i \rangle_{i=1}^n$; and let $P(\varphi) = \{\emptyset, \{a_1\}, .., \{a_n\}, .., \{a_1, a_2\}, .., \{a_1, a_2, .., a_n\}\}$ be the power set over φ, i.e., the set of possible subsets of the activities of W. An *event log* $\boldsymbol{L}$ is as a multi-set of traces, of the form $\boldsymbol{L} \in P\left(W^*\right)$.

III. Related Work

In recent years, process mining [9] has garnered increasing interest in adapting causal inference for the identification and understanding of cause-effect relationships, aiming to enhance business processes . This convergence between process mining and causal inference [8] aims to address the challenge of understanding how activities and events in business processes impact observed outcomes.

A. Causality, Interpretability and Explainability in Business Processes.

In the line of combining both domains of knowledge, some relevant works are mentioned aiming to situate causality and its role of interpretability and explainability in the diverse perspectives of business processes.

B. Fazzinga et al. [12] focus on the task of interpretability, aiming to achieve a deeper understanding and clarity in tracking low-level events in multi-process event logs II. Their main objective is to address the challenge of the lack of meaningfulness in the information contained within process activities. The authors present a novel approach where they propose an Abstract Argumentation Framework to construct a compact representation called "cas-graph" using probability scores. These scores are used to measure the probability that the associated interpretation is correct. In their research [13], they take a step further in addressing the challenge of real-time interpretation of abstraction gaps between low-level events. To achieve this, they utilize a combination of real and synthetic data sets and rely on the use of graphs as the foundation for their model. This approach enables them to provide explanations for the validity of certain interpretations. Furthermore, they validate their model by establishing a correspondence between preferred extensions in the AAFIII-A and valid interpretations of the trace II.

J. Adams et al. [14] focus their research on the detection and understanding of significant changes in business processes over time, categorized as "conceptual drifts", through the use of Granger causality [10] (*causality for time series*). Unlike standard correlation tests, the Granger approach assesses whether a variable can provide additional predictive information to forecast another variable. The concept is that a variable X *"causes"* another variable Y if the historical information of X enhances the ability to predict Y compared to using only the historical information of Y. For a comprehensive understanding of conceptual drifts, they utilize causal diagrams as they establish a causal connection between variables, thereby enabling the determination of causal direction. Some models also incorporate information regarding the direction and strength of causal relationships, thereby facilitating the comprehension of causal influence.

Bart F. A. Hompes et al. [17] posit the importance of identifying causal factors that have an impact on business processes. According to their perspective, recognizing and understanding these causal factors can be key to achieving significant improvements in business processes. By identifying and understanding the factors that influence the performance and outcomes of processes, specific measures and actions can be implemented that lead to effective and beneficial improvements in terms of efficiency, quality, and overall business performance. In their approach, the authors, as mentioned in reference [14], utilize the Granger causality model [10] to analyze performance and time-related data, generating a causal factor graph that explains process performance. The proposal applies statistical models of causality to the attributes (*which contain performance indicators*) of each activity in the business process.

B. Interpretability and Machine Learning in Business Process variables.

In recent studies, it has been observed that the interpretability of process variables has been enhanced through the use of techniques in the field of machine learning and deep learning, specifically through the interpretation of so-called *"black box"* models. This mechanism employs techniques such as visualization of relevant features, identification of relative variable importance, and generation of logical rules that describe the behavior of the model. The utilization of these interpretability techniques in the context of business processes enables an understanding of how process variables influence outcomes and provides greater confidence in decision-making based on complex models.

J. Brunk et al. [16] introduce a technique called Cause-Effect Context-Aware Dynamic Bayesian Network (CECA-DBN), which is used for predicting business processes and is sensitive to context. The main objective of this technique is to characterize the cause-effect relationships in events corresponding to process instances. Although the approach used in the CECA-DBN technique is based on and shares similarities with Bayesian networks [11], it differs in its main objective and how it is utilized to make context-sensitive predictions in the business process. While Bayesian networks rely on probabilistic inference and capture complex relationships between variables, in this case, the proposal focuses on capturing causality between events.

On the other hand, B. Wickramanayake et al. [15] focus on enhancing the interpretability and explainability of prediction models based on Deep Learning, which are often regarded as *"black boxes"*. To address this challenge, they explore the use of attention mechanisms in Long Short-Term Memory (LSTM) networks. They employ the technique of prefix tracing (*"a partial sequence of events"*) along with attention weights, which indicate the features that influenced the predictions made by the model. This combination of techniques enables them to gain a deeper understanding of how the Deep Learning model makes decisions and which aspects of the data are most relevant to the predictions made in the business process. The validation of the proposed approach is conducted using a real-world data set that is associated with loan application records. By utilizing real-world data, they are able to demonstrate the effectiveness and usefulness of their approach in a practical context.

Junya Tang et al. [18] emphasize the issues and critical importance of identifying and analyzing process bottlenecks in business processes. They propose an analysis approach based on the BTMDW-Fusion K-means data clustering, which enables determining temporal frequencies and assists the model in identifying anomalies within the processes. This approach utilizes the outcomes of data clustering to construct knowledge graphs that are processed through hyperbolic neural network models. These models conduct a multidimensional analysis of the root causes of bottlenecks from a process perspective, employing Petri nets (*"a widely used process flow modeling notation in process mining"*).

C. Causal Understanding and Counterfactual Reasoning in Business Processes.

Recently, there have been some works addressing the pursuit of a deeper understanding of the conceptualization of causality and counterfactual reasoning in the domain of business processes.

Mahraz Sadat et al. [19] underscore the significance of developing comprehensive and intricate analysis alternatives for business processes. It is proposed to apply structural equation modeling (SEM) models to the identified characteristics in the event logs related to the business process. The main objective is to utilize counterfactual reasoning to enhance the explainability in the analysis of these processes. The authors' work aims to be integrated as a plugin into the ProM process mining tool.

Qafari et al. [20] propose an alternative method to [19] by applying a five-step causal discovery process: (i) In the initial data preprocessing stage, they generate data enrichment by incorporating new features, such as performance and time values; (ii) They generate data tables with the enriched data, referred to as *"situation feature tables"*; (iii) Transformation techniques are applied to the created situation tables to obtain a set of data pairs; (iv) By utilizing the modeled data from the sets of data pairs, they construct knowledge graphs that systematically identify the structured causal relationships among the process features; Lastly, they present the strengths of the causal relationships between entities and the effects of their interventions to identify the root causes of the process.

C. Hsieh et al. [21] design an extension of DiCE called DiCE4EL (DiCE for Event Logs), which focuses on generating counterfactual explanations for predicting the next event in a process. DiCE4EL is specifically developed to address the analysis of event logs, which contain sequential information about past activities in the process. The work developed is based on the application of existing counterfactual algorithms in the Explainability of Artificial Intelligence (XAI) literature.

IV. PROPOSED TECHNIQUE

The proposal encompasses a methodical approach that is divided into four essential phases, as characterized in Fig.3. Firstly, it focuses on activities related to the field of Process Mining and encompasses two key stages in its implementation: event logging and identification of business process models.

A. Business Process Event Log

TABLE I
EVENT LOG EXTRACT

Case Id	Activity	Time-span	Feature$_1$	...	Feature$_n$
case 1	A	10.3 ms	feat.$_1$	...	feat.$_2$
case 1	B	45.2 ms	feat.$_2$	...	feat.$_3$
case 1	C	62.5 ms	feat.$_4$	...	feat.$_6$
case 2	A	72.4 ms	feat.$_5$	...	feat.$_3$
case 2	E	43.8 ms	feat.$_1$	...	feat.$_2$
...	...	...	...	...	...

As an initial step, it becomes essential to obtain the comprehensive record of all activities carried out throughout the operational flow. These records must essentially encompass the variables that constitute the pattern of linkage between the different activities, as detailed in the table I. This pursuit

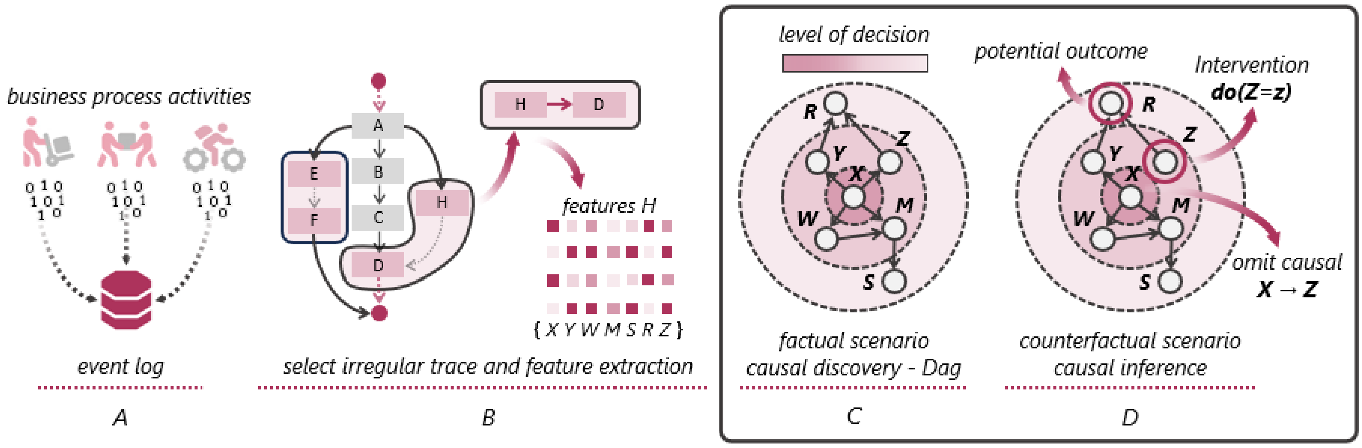

Fig. 3. The illustration of the proposed technique consists of four stages: in (**A**) and (**B**), emphasis is placed on the use of process mining techniques as an exploratory mechanism for identifying the operational reality; in (**C**), causal relationships among the variables of the business scenario of interest are identified; and finally, in (**D**), various improvement actions are evaluated and explained, conducting counterfactual analysis on the DAG II.

aims at achieving a thorough and precise understanding of the development of all actions undertaken within the pertinent process.

The event log I II, within the context of process mining, as shown in Fig.3, step **A**, is defined as a chronological and well-structured record encompassing all relevant activities and events that occur during the process's execution. Each entry in the log comprises comprehensive details regarding the executed activity, its initiation and completion times, and, occasionally, supplementary attributes associated with each event. The utilization of the **"Case Id"** field allows for precise monitoring of process execution, given that each activity is recorded with its corresponding identifier (*Trace* II). Consequently, this facilitates the analysis of action sequences, execution durations, and the detection of potential bottlenecks or irregularities.

The acquisition of an event log may vary depending on the context and the type of system or process under analysis. Many organizations automatically record activities in their enterprise systems, these systems encompass Customer Relationship Management (CRM), Supply Chain Management (SCM), and various other types of systems, which maintain event logs for operational and analytical purposes.

B. Discovery of the Business Process Model

The purpose of this stage is to obtain a comprehensive understanding of the operational reality, which, on certain occasions, may diverge from the managers' understanding and not align with the organization's established objectives. To achieve this, an analysis of event logs II is conducted to identify patterns, workflow streams, trends, and particularly bottlenecks within the business process, as illustrated in Fig.3, step **B**. This analysis is carried out through process model discovery techniques using process mining, an analytical approach that employs data mining techniques and algorithms to extract valuable information from event data.

Based on the event log II, a variety of algorithms are explored for process discovery [22]. These algorithms are evaluated in terms of their relevance regarding the level of conformity of each trace $\boldsymbol{\sigma}$ II, which is represented through $\boldsymbol{tokens}$ throughout the process model $\boldsymbol{\mathcal{M}}$. A token can be defined as an abstract symbol representing a unit of work or activity within a specific process. These tokens play a crucial role in monitoring and documenting the progress of a particular process instance as it moves through its various stages or activities.

$$f(\boldsymbol{\sigma}, \mathcal{M}) = \frac{1}{2}\left(1 - \frac{m}{c}\right) + \frac{1}{2}\left(1 - \frac{r}{p}\right) \quad (5)$$

For a process model $\boldsymbol{\mathcal{M}}$, the fitness $\boldsymbol{f}$ level of a trace $\boldsymbol{\sigma}$ 5 can be assessed by considering four pertinent variables: $(\boldsymbol{i})$ $\boldsymbol{p}$, which indicates the number of tokens produced; $(\boldsymbol{ii})$ $\boldsymbol{c}$, representing the number of tokens consumed; $(\boldsymbol{iii})$ $\boldsymbol{m}$, denoting the number of missing tokens; and $(\boldsymbol{iv})$ $\boldsymbol{r}$, corresponding to the number of remaining tokens. Consequently, the fitness measure for trace $\boldsymbol{\sigma}$ can be formulated as a function involving these variables.

C. Causal Discovery

In this stage, two approaches for causal discovery are proposed [23]. The main purpose is to obtain the Directed Acyclic Graph $\mathcal{G}$, as illustrated in Fig.3, step **C**, which serves as a source of causal knowledge among the variables corresponding to the analysis scenario of the process model obtained in Step **B**. This will allow understanding the direction, depth, and complexity of the variables under hypothetical scenarios of improvement (in this case, counterfactuals).

In this point, two approaches to causal discovery are proposed. (i) Firstly, functional causal models are employed, which rely on knowledge about the functional forms and characteristics of distributions governing relationships between variables. These models use the so-called Additive Noise Model and accurately identify causal directions within a specific dataset. An example of an algorithm belonging to this family is Direct Linear Non-Gaussian Acyclic Model **DirectLiNGAM** [24]. (ii) Secondly, we find constraint-based models, also known as independence-based approaches. These models aim to unravel the causal structure of the data by exploiting independence between three elementary graphical models. A representative algorithm of this family is **PC** [25]. This method begins with a graph containing edges between all variables, forming a fully connected network. Subsequently, edges representing relationships of unconditional independence between variables are eliminated. Furthermore, iterations are performed over all possible pairs of remaining variables, and edges that are not necessary under certain conditional associativity conditions are removed.

D. Counterfactual Explainability

This stage is only possible if we have the identification of the causal relationship and its interpretation through the mechanism of a DAG $\mathcal{G}$, in which explainability can be developed mainly for relationships as illustrated in Fig.4

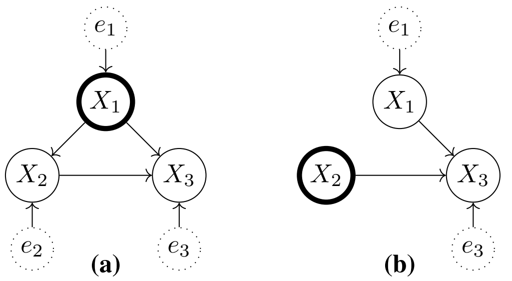

Fig. 4. **(a)** Is the causal relationship with their respective additive noises e_i, according to **DirectLiNGAM** [24], and **(B)** is the scenario of interest to provide counterfactual explainability.

In the context of the business process, the causal relationships among variables provide insights into the direction and extent of potential decision-making. Specifically, X_1, from the standpoint of causal inference, functions as a confounding variable: a supplementary factor that can influence both the independent variable (cause) X_2 and the dependent variable (effect) X_3. Therefore, X_1 must be properly controlled, as it can create a spurious association between the cause and the effect, leading to erroneous conclusions about the causal relationship between them.

The Step **D** in Fig.3 illustrates a systematic and controlled approach to developing the contractual scenario. For instance, let us assume that the variable X_3 is of particular interest for implementing an improvement action within the business context. Essentially, the aim is to explore the effects of intervening in variable X_2 with the purpose of achieving an outcome aligned with the business requirements in X_3. This intervention is described by the conditional distribution $p(X_3 \mid do(X_2 = x))$.

To ensure the validity of the causal analysis, it is imperative to maintain the stability of the causal graph in all its connections, except for the causal dependencies (parent nodes) that influence the variable under intervention, in this case, X_2. This precaution ensures the accuracy and reliability of the inferences. Fig.4.$\boldsymbol{b}$ provides a visual representation of this process, demonstrating the exclusion of the parent nodes when conducting the intervention.

V. Results

With the purpose of applying and validating the proposed technique IV, a set of authentic data from the industry has been employed. This dataset describes various activities and the operational reality related to the conception, adaptation, and distribution process of payment media cards. It is relevant to highlight that the company responsible for the management and production of these cards has a nationwide scope and is primarily dedicated to meeting the demands of banking institutions and the retail sector. In order to ensure confidentiality, the identity of this company has been preserved in the current analysis.

A. Event Log exploration and Process Model Discovery

The analyzed Event Log comprises a total of 18 activities and 128,544 process instances (number of cards), corresponding to 2,088,347 events and 4 process variants.

TABLE II
Business Process Activities

Activities
Exploitation, Embossing Legal Documentation Printing
Legal Documentation Printing, Plastic Legal Embossing
Documentary-Type Printing, Documentary-Type Embossing
Compiling Reception, Assembly of Packages, Kits/Sets Laminating
Kits/Sets Vault, Vault Requests, Reception Classification
Kit Delivery Classification, Payroll Classification
Distribution to Customers, Kit Control, Kit Delivery Classification
Kit Control

Furthermore, la Fig.5 presents a perspective of the times at which bottlenecks occur, such as the execution times and waiting times between activities. The boxes indicating the duration of each process activity take on a lighter shade to signify shorter durations, while the darker boxes indicate longer durations. This representation resembles a calorimetric scale, where higher values are depicted in red and lower values in fainter tones.

The perspective on execution times presented in Fig.5 is based on the use of the median as a measure of central tendency. The choice of the median as a statistic is justified because this measure tends to more accurately highlight points in the process that experience greater slowness. Additionally, the median is characterized by its robustness to atypical values, making it a suitable option for mitigating the impact of extreme observations in the analysis of execution times.

The foregoing (i) indicates that activities related to collating and printing exhibit significantly high execution time intervals. For instance, the activity of "Documentary-type Printing" has a median execution time of 2 hours per unit of Card Kit. Additionally, this activity presents non-established variations in the process execution standard, resulting in delays

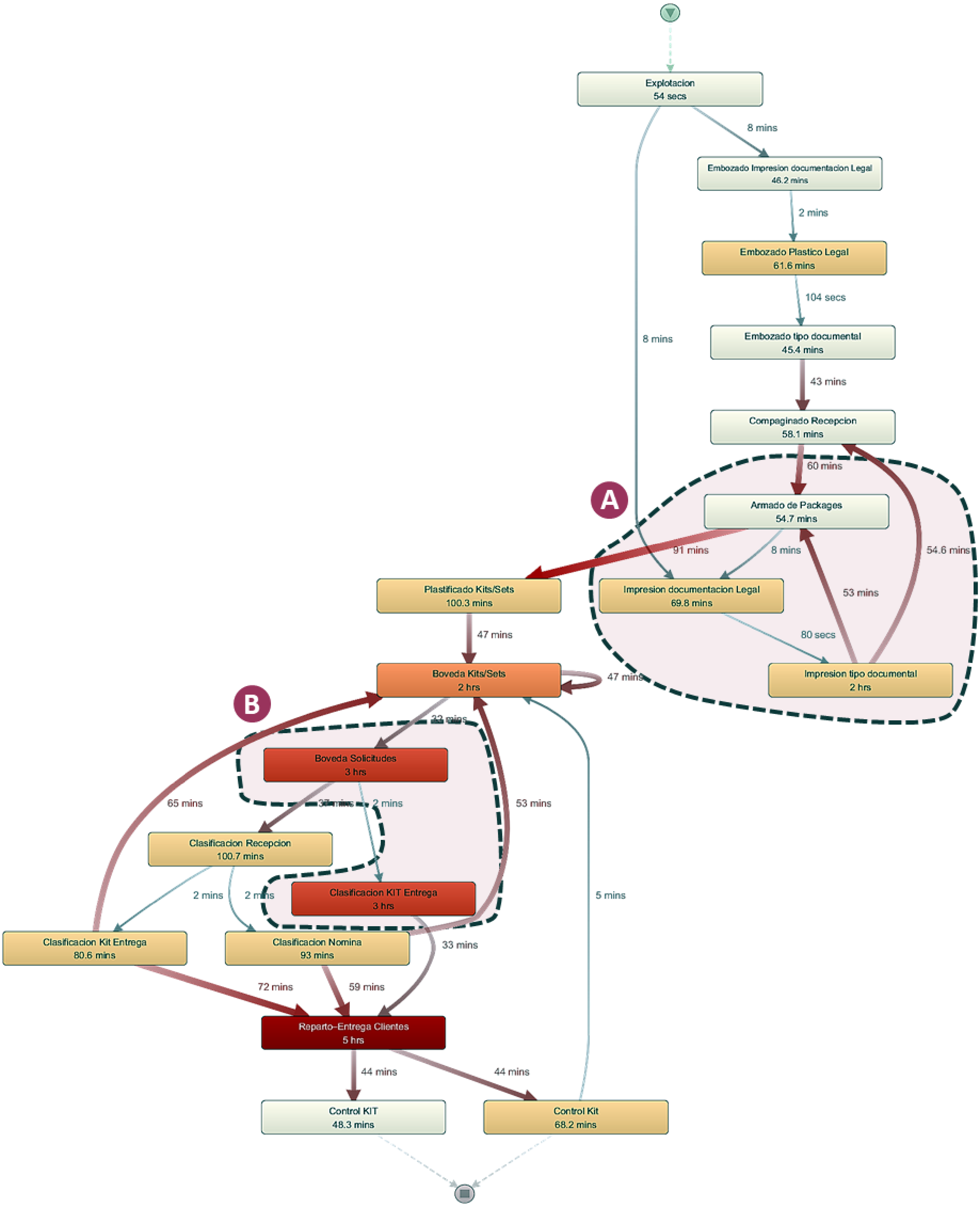

Fig. 5. The derived process model originates from the process mining tool named ”Disco”, which is grounded on an extension of the ”Fuzzy Miner” algorithm. It stands out for achieving suitable fit levels in the face of cyclical flows between activities. In this instance, both the paths and the activities identified in the event log are fully represented, accounting for 100%.

in the elaboration of Card Kits. (ii) It is at this juncture where activities that consume the most execution time throughout the process stand out, specifically those related to "Vault" and "Classification". Activities such as "Vault Requests" and "Kit Delivery Classification" reach median execution times of 3 hours.

B. Selection of the Scenario of interest by the Business.

According to the experience of process managers and analysts, it is of utmost importance to address the causal analysis present in the scenario of Fiq.5, scenario **A**. This is because printing activities involve both manual and automated tasks. However, the implementation of resources in these areas has led to the development of flow variations among activities that are not visible in the operational reality for the managers. The present analysis focuses on the link between the tasks known as "Legal Documentation Printing" and "Assembly of Packages." This sequence of activities is associated with the return of kits in process that exhibit non-validated attributes. Despite the familiarity with these scenarios, the process stakeholders seek a comprehensive understanding of the causal relationships that manifest in the "Legal Documentation Printing" activity. Specifically, they aim to discern the underlying reasons for the waiting time between both activities, while excluding the possibility that such delays originate from the "Assembly of Packages" task. This justification arises from the absence of temporal constraints concerning the input of work instances into the said activity.

C. Causal Model based on the variables of the Scenario of Interest.

Considering the peculiarity of the industry in which the company is situated and the business process focused on in this paper, characterized by its marked competitiveness, the company has managed to forge a significant competitive advantage over its competitors over time. In order to safeguard the identity of this entity, only the scope of the variables deemed relevant for the task of causal discovery is mentioned.

The variables involved in the analysis have scope within the domain of role utilization R_i (users with defined tasks in the process), use of inputs and materials M_i, production quantities Q_i, and T_i times associated with manual procedures. We begin by identifying the causal relationships. The general idea behind **DirectLiNGAM** is as follows: we assume that we have variables with additive noise, and the noise is independent of the cause, i.e., $Y = f(X_i) + \epsilon$ and $X_i \perp\!\!\!\perp \epsilon$.

Hence, this condition is fulfilled when evaluating the variables R_1 and R_2 as causal factors for T_1, resulting in a causal relationship with T_3.

A preliminary review of the identified DAGs indicates the existence of a causal relationship between the materials M_i and the procedure times within the scenario. This implies the necessity of conducting a review and potential improvement of the levels established by the process. Although the materials and supplies are available, there could be a situation where

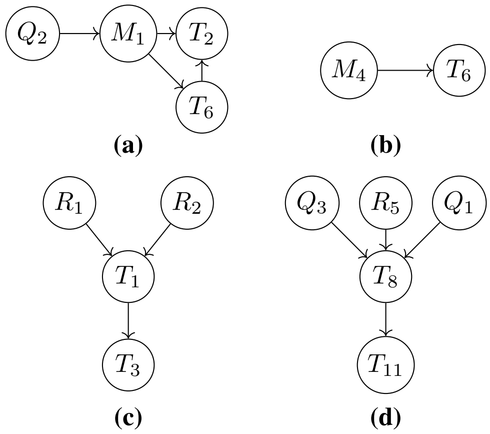

Fig. 6. Identified DAGs represent causal relationships among various variables that have domains in diverse scopes within the activities.

alert events are not aligned with demand events, as evidenced by the causal impact from $\boldsymbol{Q_2}$ (Fig.6.**a**).

D. Counterfactual explainability and its potential effects..

The following counterfactual development will be conducted on an identified DAG, as mentioned in the preliminary DAG (Fig.6**a**), posing the following question: What effect would having a greater quantity of materials and inputs to address times T_1 and T_2 have?

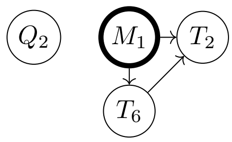

Fig. 7. Counterfactual development; an intervention is performed on M_1.

Indeed, there exists a causal relationship between the variables. It is feasible to intervene in the associated timing of these variables. The direct causal relationship between the variables can be approached as an unbiased linear impact, accounting for additive noise with independence between the causal factor in $T_2 = f(M_1) + \epsilon$ and $M_1 \perp\!\!\!\perp \epsilon$.

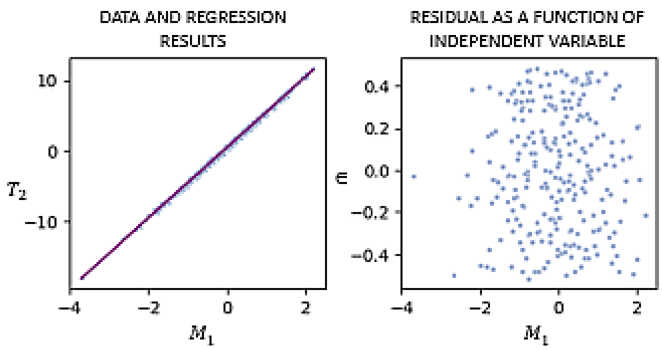

Fig. 8. $T_2 = f(M_1) + \epsilon$ and $M_1 \perp\!\!\!\perp \epsilon$.

VI. Conclusions and Future Work

Causal graphs are a robust tool for understanding causal relationships in scenarios that require identifying complex cause-and-effect relationships in an industrial context. Additionally, they provide a causal order topology, which exposes the level of decisions that the manager must evaluate and the causal impact on the process. The development of counterfactuals is a solid tool for analyzing assumptions about observed data.

This implies taking earlier actions and evaluating possible improvement actions more focused on the process. In this paper, we address runtime scenarios that do not align with process requirements, but it is also promising to address other irregular scenarios, which can be complemented with unsupervised learning techniques to identify various types of anomalies.

The proposed technique in this paper is capable of providing answers to the questions posed (Q_1I-B, Q_2I-B, and Q_3I-B) since it was possible to identify key variables based on the topology in the Directed Acyclic Graphs (DAGs). By considering the direction and branching of a decision regarding a causal variable, it also allowed determining which mechanisms are exposed for developing interventions.

The identified weaknesses are associated with the presence of variables for which causal relationships were not identified. The reason for this absence may vary, or it simply may not have been possible to address it. It is important to consider that organizations may not record all process characteristics in their information systems, which can affect the ability to establish complete causal relationships. Before conducting causal analysis, it is crucial to consider this aspect to set reasonable expectations for the managers. By understanding the limitations of the available data and the possible unaddressed variables, more accurate interpretations can be made, and erroneous conclusions can be avoided. This ensures a more informed and effective decision-making process within the context of causal research.

This paper stands out from related works in the field of process mining as, to the best of our knowledge, causal functional relationships have not been evaluated in the literature. In this study, we address a feature space that, while outside the scope of the process model, is important to consider. This is because underlying causals cannot be identified from the process model itself. Instead, the model will reflect causal effects that must be explored in an exogenous feature space. By considering this exogenous feature space, the study aims to understand and analyze causal relationships that may be present but are not directly captured by the process model. In doing so, it broadens the focus to gain a better understanding of the causal connections influencing the process and allows for a more comprehensive approach in identifying and evaluating relevant causal functional relationships. It is important to emphasize that the inclusion of this analysis in an exogenous feature space can significantly enrich the understanding of the studied phenomena and contribute to a more holistic approach in the analysis and optimization of processes.

An emerging approach in process mining is the utilization of object-centric models, which tackle the more intricate operational reality of processes that interact with each other within an organization. This poses challenges in determining causal relationships and necessitates addressing counterfactual inference within an even more complex topology of variables.

Acknowledgements

This work has been partially supported by ANID under grant ANID PIA/APOYO AFB220004 (CCTVal).

References

[1] Mona von Rosing, Henrik von Scheel, Justin Tomlinson, Victor Abele, Kenneth D. Teske, Michael D. Tisdel, Business Process Management Alignment, Editor(s): Mark von Rosing, August-Wilhelm Scheer, Henrik von Scheel, The Complete Business Process Handbook,Morgan Kaufmann,Pages 649-660, 2015.

[2] Kaouther Mehdouani, Nesrine Missaoui, Sonia Ayachi Ghannouchi, An approach for Business Process Improvement Based on Simulation Technique, Procedia Computer Science, Volume 164, 2019.

[3] N.C. Proudlove, S. Bisogno, B.S.S. Onggo, A. Calabrese, N. Levialdi Ghiron, Towards fully-facilitated discrete event simulation modelling: Addressing the model coding stage, European Journal of Operational Research, Pages 583-595 (2017).

[4] Pierluigi Zerbino, Alessandro Stefanini, Davide Aloini,Process Science in Action: A Literature Review on Process Mining in Business Management,Technological Forecasting and Social Change, Volume 172, 2021

[5] Yao, Liuyi, et al. "A survey on causal inference." ACM Transactions on Knowledge Discovery from Data (TKDD) 15.5 (2021): 1-46.

[6] Alessio Zanga, Elif Ozkirimli, Fabio Stella, A Survey on Causal Discovery: Theory and Practice, International Journal of Approximate Reasoning, Volume 151, 2022.

[7] Yang Zhao, Qing Liu, Causal ML: Python package for causal inference machine learning, SoftwareX,Volume 21, 2023.

[8] Kun Kuang, Lian Li, Zhi Geng, Lei Xu, Kun Zhang, Beishui Liao, Huaxin Huang, Peng Ding, Wang Miao, Zhichao Jiang, Causal Inference, Engineering, Volume 6, Issue 3, 2020.

[9] dos Santos Garcia, C., Meincheim, A., Junior, E. R. F., Dallagassa, M. R., Sato, D. M. V., Carvalho, D. R., ...& Scalabrin, E. E. Process mining techniques and applications–Asystematic mapping study. Expert Systems with Applications, 133, 260-295, (2019).

[10] Shojaie, A., Fox, E. B. Granger causality: A review and recent advances. Annual Review of Statistics and Its Application, 9, 289-319. 2022.

[11] Kitson, N. K., Constantinou, A. C., Guo, Z., Liu, Y., Chobtham, K. A survey of Bayesian Network structure learning. Artificial Intelligence Review, 1-94. (2023).

[12] Bettina Fazzinga, Sergio Flesca, Filippo Furfaro, Elio Masciari, Luigi Pontieri, Efficiently interpreting traces of low level events in business process logs, Information Systems, Volume 73, Pages 1-24, 2018.
[13] Bettina Fazzinga, Sergio Flesca, Filippo Furfaro, Luigi Pontieri, Process Mining meets argumentation: Explainable interpretations of low-level event logs via abstract argumentation, Information Systems, Volume 107, 2022.
[14] Jan Niklas Adams, Sebastiaan J. van Zelst, Thomas Rose, Wil M.P. van der Aalst, Explainable concept drift in process mining, Information Systems, Volume 114, 2023.
[15] Bemali Wickramanayake, Zhipeng He, Chun Ouyang, Catarina Moreira, Yue Xu, Renuka Sindhgatta, Building interpretable models for business process prediction using shared and specialised attention mechanisms, Knowledge-Based Systems, Volume 248, 2022.
[16] Jens Brunk, Matthias Stierle, Leon Papke, Kate Revoredo, Martin Matzner, Jörg Becker, Cause vs. effect in context-sensitive prediction of business process instances, Information Systems, Volume 95, 2021.
[17] Hompes, B.F.A., Maaradji, A., La Rosa, M., Dumas, M., Buijs, J.C.A.M., van der Aalst, W.M.P. Discovering Causal Factors Explaining Business Process Performance Variation. In: Dubois, E., Pohl, K. (eds) Advanced Information Systems Engineering. CAiSE 2017. Lecture Notes in Computer Science(), vol 10253. Springer, 2017.
[18] Junya Tang, Ying Liu, Kuo-yi Lin, Li Li, Process bottlenecks identification and its root cause analysis using fusion-based clustering and knowledge graph, Advanced Engineering Informatics, Volume 55, 2023.
[19] Qafari, M.S., van der Aalst, W.M.P. Case Level Counterfactual Reasoning in Process Mining. In: Nurcan, S., Korthaus, A. (eds) Intelligent Information Systems. CAiSE 2021. Lecture Notes in Business Information Processing, vol 424. Springer, 2021.
[20] Qafari, M.S., van der Aalst, W.M.P. Feature recommendation for structural equation model discovery in process mining. Prog Artif Intell (2022)
[21] C. Hsieh, C. Moreira y C. Ouyang, "DiCE4EL: Interpreting Process Predictions using a Milestone-Aware Counterfactual Approach", 2021 3rd International Conference on Process Mining (ICPM), Eindhoven, Países Bajos, pp. 88-95, 2021.
[22] Dianmin Yue, Xiaodan Wu, Haiyan Wang and Junbo Bai, "A review of process mining algorithms," 2011 International Conference on Business Management and Electronic Information, Guangzhou, 2011.
[23] Alessio Zanga, Elif Ozkirimli, Fabio Stella, A Survey on Causal Discovery: Theory and Practice, International Journal of Approximate Reasoning, Volume 151, Pages 101-129, (2022)
[24] Shimizu, S., Inazumi, T., Sogawa, Y., Hyvarinen, A., Kawahara, Y., Washio, T., ... & Hoyer, P. . DirectLiNGAM: A direct method for learning a linear non-Gaussian structural equation model. Journal of Machine Learning Research-JMLR, 12(Apr), (2011).
[25] Spirtes, P., Glymour, C., and Scheines, R. Causation, Prediction, and Search. The MIT Press, Cambride, MA, 2nd edition. (2000)

4.3 Causal Organizational Mining in Software Engineering: Evaluating Improvement Strategies in Development Team Dynamics Process

After addressing in the previous articles the structural identification of variants and counterfactual explainability over the process flow (**L**$_2$-**L**$_3$), the third article broadens the causal framework towards the organizational perspective. It analyzes interaction topologies between roles as an explicit component of the causal model and integrates their effect on temporal performance, quantifying at the interventional level of the hierarchy (**L**$_2$) how patterns of delegation and collaboration behave as root causes of inefficiencies and enable structural interventions at the team level (**SO.1**, **SO.3**; **RQ.1**, **RQ.3**). In this way, it extends the organizational mining literature, which typically describes collaboration networks, by proposing that such networks be interpreted and used as causal mechanisms over temporal performance.

Publication Summary

Title:	*Causal Organizational Mining in Software Engineering: Evaluating Improvement Strategies in Development Team Dynamics Process*
Authors:	Fernando Montoya, Cristian Beltran and Hernán Astudillo
Conference:	50th Latin American Computer Conference (CLEI)
Publisher:	IEEE *Xplore*
Year:	2024
Doi:	https://doi.org/10.1109/CLEI64178.2024.10700073

Causal Organizational Mining in Software Engineering: Evaluating Improvement Strategies in Development Team Dynamics

Fernando Montoya
Nexus Payment Systems Spa.
Santiago, Chile
&
Fundación Instituto Profesional Duoc UC
Santiago, Chile
&
U. Técnica Federico Santa María
Santiago, Chile
fernando.montoya@usm.cl

Cristian Beltran
Fundación Instituto Profesional Duoc UC
Santiago, Chile
cbeltranf@duoc.cl

Hernán Astudillo
ITiSB - Universidad Andrés Bello
Viña del Mar, Chile
&
U. Técnica Federico Santa María
Santiago, Chile
hernan@acm.org

***Abstract*—**
In the context of software engineering, analyzing causality in the dynamics of development teams is essential for optimizing performance. Causality involves understanding how certain factors (organizational structures, individual interactions) directly influence the team's performance, allowing for the identification and implementation of effective improvements in development processes. Identifying and understanding the underlying causes of heterogeneous effects in team dynamics is essential for improving collaboration and productivity. The lack of specific methodologies to explore these causal relationships in complex organizational settings limits the ability of project leaders to implement effective changes. This article describes a method that uses causal inference in organizational mining to assess software development teams. Data are analyzed to identify interactions and causal factors affecting role dynamics, and causal inference techniques are employed to evaluate the effects of improvement actions. The effectiveness of this approach was confirmed with a pilot software development team at a Chilean payment processing company. By employing causal organizational analysis methods, managers were able to select more focused strategies, based on a deep and detailed understanding of the underlying causal dynamics. This work contributes to the field of software engineering by introducing a structured and causal approach to analyze team dynamics and providing project managers with tools to address the underlying factors that hinder team effectiveness.

***Index Terms*—organizational mining, causal inference, software engineering, business process mining.**

I. Introduction

In the current and dynamic software development environment, marked by constant technological advances, maintaining operational stability is a key challenge for development teams. This stability depends not only on the tools and technologies used but also crucially on the synergy and dynamics among team members. It is essential to balance the participation of each member and effectively manage the variables that affect role interactions, allowing contributions according to

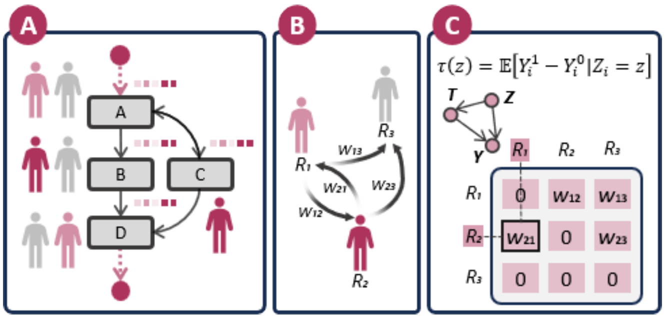

Fig. 1. Characterization of team dynamics in software development, weighting of roles in procedural interaction, and modeling of topological causality to estimate heterogeneous effects.

their skills, especially during periods of significant change, such as adjustments in project scale. The operational variables that underpin the process have a direct impact on agility, automation, and communication among team members [1]. Changes in these variables can arise, among other factors, from software requirements that are incomplete, contradictory, or prone to frequent modifications [2]. These changes in requirements disrupt the operational dynamics, affecting both the workflow and the coordination among team members. Such variations require continuous adjustments, posing a significant challenge for the evaluation and selection of strategies that enhance both the efficiency and operational cohesion of the team.

Faced with these challenges, there arises a need to adopt methods that incorporate a causal perspective to thoroughly understand the relationships between cause-and-effect variables. However, this approach encounters significant difficulties in the absence of expert domain knowledge, especially when the understanding of the operational causes within the

979-8-3315-4097-5/24/$31.00 ©2024 IEEE

development team is limited, often leading to decisions biased more toward correlations than causality [3]. Therefore, it is crucial to deepen the understanding of the operational and causal dynamics of the roles to ensure precise and adaptable management, addressing the diversity of effects that may arise from operational improvement actions.

A. Organizational Perspective

Fig.1, in Phase A, characterizes the modeling of the software development process flow from two perspectives: organizational and causal. From the organizational perspective, it highlights how team members interact with variables instantiated by the process, such as the number of modules to be developed for an interface or the number of functionalities according to design or architectural requirements [4], among others. This interaction can be analyzed through organizational mining [6], a methodology that studies work patterns using business records to understand how teams collaborate and improve task allocation and management. By focusing on creating models based on real work activities, organizational mining helps companies organize their teams more efficiently, ensuring that the right people work on the right tasks. This reflects a direct integration of theory and practice in the continuous improvement of processes in the field of software development.

Section B describes the topology of the interaction network among the roles in the process. From an organizational perspective, different instances of the development process can generate a variety of interactions among team members. In this context, the goal is to identify and understand topologies such as: (i) work delegation, (ii) subcontracting, (iii) task reassignment, (iv) team collaboration, and (v) execution of similar tasks [6][7]. Each topology develops based on operational variables, which are defined by specific tasks. This reflects a causal dynamic, whether planned or emergent, that arises from interactions among team roles. This causal relationship demonstrates how actions and decisions within the team can lead to different outcomes depending on the established interaction structure.

B. Causal Perspective

Section C introduces an adjacency matrix M [8] to detail and visualize the interactions among team members. This matrix, structured as a two-dimensional array, assigns each row and column to a specific role within the team, such as R_2 and R_1. The elements of the matrix, algebraically represented as w_{ij}, indicate the presence or intensity of the interaction between the role of row i and the role of column j. A non-zero value in w_{ij} suggests a direct interaction between these roles, providing managers with a clear and quantifiable view of the work topologies.

Interconnections often contain operational variables that support a causal structure [10] not always aligned with the knowledge of domain experts, a structure that is crucial for the development of effective interventions in the process. Additionally, interventions in an operational process can have varied effects due to covariates [12], as illustrated by the Directed Acyclic Graph (DAG) [13] [14] in Section C. It is essential to consider confounding variables such as Z, which are linked to outcomes and directly affect them, to understand the variability in the effects of interventions. Interconnections often contain operational variables that sustain a causal structure not always aligned with the domain experts' knowledge, a structure that is vital to understand in order to develop process improvement interventions.

Taking the software development process as an example, it is crucial to understand how variables such as team collaboration (T) and client satisfaction (Y) are influenced by project management practices (Z). If the influence of Z is not considered, managers might implement improvements that do not effectively solve issues related to collaboration or client satisfaction, due to the unrecognized impact of these management practices.

C. Challenge and Questions

This work focuses on identifying causal relationships between variables aimed at improving operational performance in the interaction activities among team roles within a development process. To this end, it proposes a novel approach that integrates organizational mining techniques, a subdomain of process mining [15], with causal inference methods [16].

The causal approach proposed in this work has not yet been explored in role interactions within development teams. Existing literature has addressed team member connections through correlational or probabilistic analysis, focusing on demographic and sociocultural variables. However, these methods do not confirm causality between variables, thereby limiting a deep understanding of team dynamics and their impact on operational performance.

- $\boldsymbol{Q_1}$: *"What is the underlying causal structure in the operational interactions among the team roles?".*
  The main task to be addressed is the identification and exploration of the causal structure, aiming to understand in which direction the variables are positioned to establish a treatment condition (causal variable) and outcome (effect variable). This involves determining in which direction and space of possibilities decisions on improvement actions should be made and how these branch out within the causal network.
- $\boldsymbol{Q_2}$: Once the causal structure has been identified, *"which instances within the topological space are most likely to respond favorably to improvement interventions?".*
  Based on historical and observational data, the main task for this question is to conduct a counterfactual analysis on the causal network. In this context, hypotheses can be formulated to assess what would have happened if a certain action had been taken.

The remainder of this paper is structured as follows: Section II causal effect; Section III surveys related work; Section IV proposed technique; Section V results; Section V conclusions;

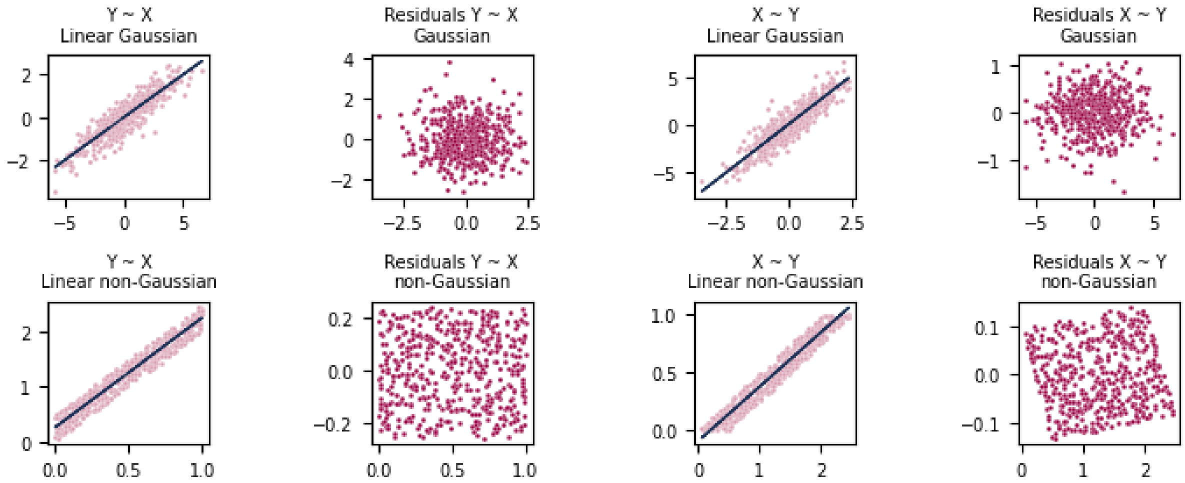

Fig. 2. The illustration reveals how the plotted line synthesizes the relationship between the study variables according to the optimal estimation of the employed causal model, which is crucial in the analysis to uncover causal relationships. It is noted that X is causal to Y, denoted as $X \rightarrow Y$, when the residual ϵ is not Gaussian and $Y = f(X) + \epsilon$, where $X \perp\!\!\!\perp \epsilon$, as reflected in the lower graphs.

II. Causal Effect

To determine if a variable X has a causal effect on another variable Y ($X \rightarrow Y$), it is crucial to confirm the existence of a causal relationship. The discipline of causal discovery [9], which investigates this concept, encompasses a range of methodologies designed to reveal information about causal mechanisms using observational data. Essentially, algorithms in this domain aim to decode the causal framework underlying the data generation process.

A. *Causal Discovery*

X and Y could be related in such a way that one causes changes in the other, such that this relationship is expressed through a linear function. Functional causal discovery algorithms [31] focus on identifying whether there is a causal connection governed by a specific function. Fig.2 evaluates whether X causally affects Y in two different scenarios: the upper graphs show residuals with a Gaussian distribution for both $Y \sim X$ and $X \sim Y$; the same occurs in the lower graphs, but in this case, the residuals follow a non-Gaussian distribution.

The main idea of functional causal discovery algorithms is to find a causal structure in which the exogenous variables to the function $Y = f(X) + \epsilon$ are independent of the noise associated with the causal function. In terms of Fig.2, the lower graphs reveal that under the scenario where the residuals do not exhibit a Gaussian distribution, it can be asserted that X has a causal relationship with Y, given that X is independent of the noise ϵ, meaning $X \perp\!\!\!\perp \epsilon$.

B. *Causal Inference*

Once the causal structure is identified, it is possible to develop inferences that explain the influence of a set of variables on others. Estimating the causal effect $\tau = \mathbb{E}\left[Y(1) - Y(0)\right]$, which measures the expected difference in outcomes between treated and control groups, facilitates understanding of causal hierarchy, intervention reasoning, and counterfactuals. In the first case, we explore what would happen by intervening in a variable; in the second, what would have occurred if a specific action had been taken. Although they may seem similar, they differ in their assumptions and approaches. Mechanisms that allow for the evaluation of improvement strategies in a specific scenario using causal reasoning.

III. Related Work

According to the literature review, the application of organizational mining techniques combined with the domain of causal inference to assess operability in development teams has not been identified. This absence highlights a significant opportunity to investigate innovative perspectives in the field, especially considering that existing studies have predominantly focused on the use of quality diagrams and probabilistic graphs.

A. *Cause-Effect Diagrams*

To characterize cause-and-effect scenarios, Haneen et al. [24], Fausto et al. [25], and Nicolli et al. [26] focus on using cause-and-effect quality diagrams, such as the Ishikawa diagram, to address various issues in software development. Haneen et al. [24] analyze how risk factors like differences in time zones, languages, and cultural practices influence the performance of development teams. Fausto et al. [25] apply a similar approach to identify challenges in developing new products, highlighting how unexpected events and uncertainties can impact the team. Meanwhile, Nicolli et al. [26] use Ishikawa diagrams with probabilities to manage technical debt, providing teams with tools to understand and control causes and their consequences.

In relation to other factors of interest, such as requirements evaluation, Woo et al. [27] propose an automatic cause-effect tool as an intermediate model to simplify complex requirements and integrate them with two units aimed at eliminating redundancy in the requirements.

B. Causal Networks

Prateeti et al. [28] employ graph models, such as Structural Equation Modeling (SEM), to explore the impact of geographical and cultural challenges on the collaboration of global software development teams, enabling a detailed analysis of the connections between variables in complex contexts. In a complementary approach, Laura-Diana et al. [29] use Bayesian networks to refine effort estimations in agile projects, focusing their analysis on how the quality of teamwork and the characteristics of user stories, affected by changes in requirements and team dynamics, influence project success. Similarly, Arthur et al. [30] develop a model based on Bayesian networks, using studies and expert advice to evaluate and improve the quality and dynamics of teamwork in agile projects.

These authors conclude that traditional methods, such as structural equation modeling and radar charts are ineffective for predicting or diagnosing quality issues in teamwork, highlighting the need for more sophisticated approaches.

IV. Proposed Method

The proposal encompasses a methodical approach that is broken down into four essential phases, as shown in Fig.3. Initially, stages A and B focus on activities related to the field of Process Mining. Stages C and D cover the domains of discovery and causal inference.

A. Log events

The goal of this stage is to extract a development process flow using process discovery techniques and organizational mining on event logs, which can be extracted from domain information systems, such as Azure DevOps Board. This allows for the construction of an organizational topology, as described in Section I-A. The event log of the process can be characterized according to Table.I. The importance of the event log lies in the fact that capturing the dynamics among the roles of the development team is based on an executed reality and is not limited to a modeled or typified reality by the existing documentation in the organization. This allows for the revealing of pathways between activities and events that are not visible to managers, enabling a broader analytical space that can have a direct causal relationship on operability. The techniques applied in this phase belong to the domain of process mining [22].

TABLE I
Event Log Extract

Case Id	**Activity**	**Time-span**	**Role$_1$**	**...**	**Feature$_n$**
case 1	A	10.3 ms	role.$_1$	...	feat.$_2$
case 1	B	45.2 ms	role.$_2$	...	feat.$_3$
case 1	C	62.5 ms	role.$_4$	...	feat.$_6$
case 2	A	72.4 ms	role.$_5$	...	feat.$_3$
case 2	E	43.8 ms	role.$_1$	...	feat.$_2$
...	...	...	...	...	...

The structure of Table.I describes the data necessary to implement the proposed method. Each instance that moves through the activity flow must have a unique Id that identifies the events carried out at each stage, the name of the activities, execution times, and operational variables. These elements are crucial for recognizing the actual flow executed. Regarding the *"role"* attribute, it can be identified by features such as the *"executor's name"*, *"the workstation"*, or another identifier, facilitating the identification of the interaction network among the process participants.

B. Interaction Topology

At this stage, various data are analyzed, including case identifiers and role columns mentioned in Table.I. From this data, an adjacency matrix is extracted, as shown in Fig.3.B, representing the interactions among team members. Within the context of a development process, each role performs its tasks following a predefined activity flow. This dynamic results in the interaction structure detailed in Section I-A.

The event log L acts as a repository of activities, represented by sequences $\langle a_1, \ldots, a_n \rangle$, which correspond to a trace $\sigma \in L$. These traces capture the progression of events, gathering

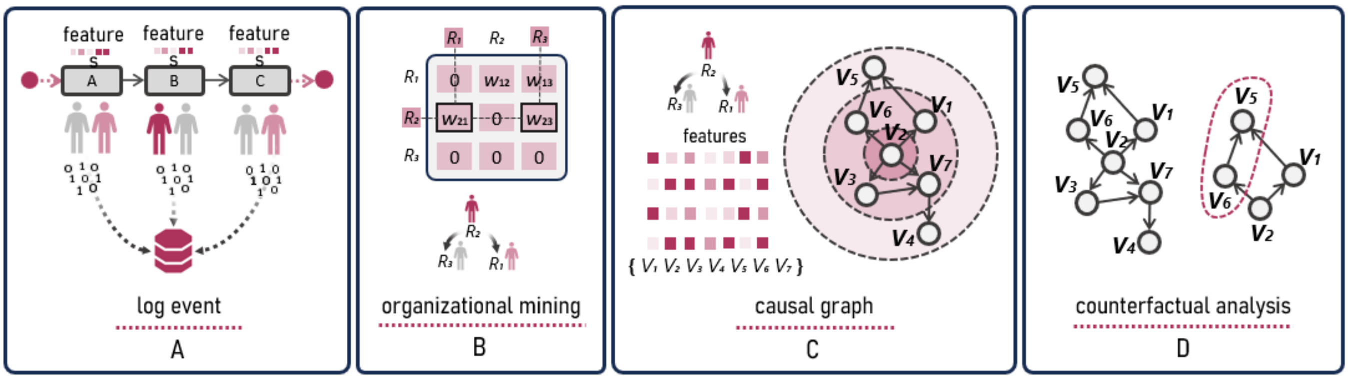

Fig. 3. Illustration of the method proposed in this work, which integrates organizational mining and causal inference techniques within the context of processes and interactions in development teams.

information about the interactions of different roles, with each activity associated with specific roles such as $role_1$, $role_2$, and $role_5$. L essentially becomes a set of interactions, intertwining the roles and actions involved in the flow of the development process.

where $L = \left[\left\langle \mathrm{a}^{role_1}, \mathrm{b}^{role_2}, \mathrm{c}^{role_2} \right\rangle, \left\langle \mathrm{a}^{role_5}, \mathrm{e}^{role_1} \right\rangle, ...\right]$. In which a topological search of interaction can be performed, such as the work delegation I-A. For example, by examining the delegation of work among roles, a topology that can be extracted from Algorithm.1, operational weaknesses can be identified. This analysis provides information about user interactions, allowing conclusions to be drawn regarding the distribution of workload and key participants in the process.

Algorithm 1 work delegation

Let M be a square matrix of size $|W|^2$
Initialize $M_{ij} \leftarrow 0$ for every position (i, j)
for each Case id in L **do**
    for each consecutive Role transition $R_i \rightarrow R_j$ in that Case id **do**
        $M_{ij} \leftarrow M_{ij} + 1$
    end for
end for

Let's consider W as the set of unique users recorded in an event log, and $|W|$ as the cardinality of this set. Given Fig.1, the matrix M is of size $|W|^2$, where $|W| = 3$, as it contains R_1, R_2, R_3, and is two-dimensional to identify interactions between roles in $|W|^2$. In this context, we are looking for a matrix that acts as a function $\mathrm{f} : \mathrm{W} \times \mathrm{W} \rightarrow \mathbb{N}_0$, which provides the number of times each possible transition between users in W has been observed in the event log.

The matrix M has dimensions $|W|^2$ and must be initialized with zeros. As we progress through the event log L, each time we observe a transition $R_i \rightarrow R_j$ in some case Id, we increment the value at the corresponding position (i, j) in the matrix M.

C. Causal Graph, Treatment Variable and Covariates.

Once the working topology of base B has been identified, the operational variables related to the Id cases connected in the instances of the work network are extracted. Upon these variables, causal discovery is applied, which underlies the working topology. The main purpose of this task is to identify the set of outcome variables (effects) that may have a causal relationship with the treatment variables (causals). The main task at this stage is to identify the structural functions that represent the causal relationship. For instance, if $X \rightarrow Y$ and $Z \rightarrow Y$, we aim to find the causal functions $Y = f(X) + \boldsymbol{\epsilon_{xy}}$ and $Y = f(Z) + \boldsymbol{\epsilon_{zy}}$, enabling managers to draw inferences. In such a case, X and Z are potential treatment variables, while Y is the affected variable. However, if in the same structure it holds that $Z \rightarrow Y$, Z becomes a covariate and a confounding variable. This variable must be taken into account in the causal effect exerted by X on Y, as Z will influence the action of X.

$$P\left(X_1, \ldots, X_n\right) = \prod_{i=1}^{n} P\left(X_i \mid \mathbf{pa}\boldsymbol{\mathcal{G}_i}\right) \tag{1}$$

As a mechanism for interpretability, a directed acyclic graph $\mathcal{G}$ is utilized, denoted according to 1. Where $\mathbf{pa}\boldsymbol{\mathcal{G}_i}$ represents the parent nodes (*causal*) of the variable node X_i (*effect*) in the directed acyclic graph $\boldsymbol{\mathcal{G}}$ (DAG). This graph can be validated through the conditional independences among the variables X_i, known as local Markov conditions (LMC) [23]. To do so, it is verified within a possible space of combinations: $\boldsymbol{(i)}$ Chain, $Z \rightarrow X \rightarrow Y$; $\boldsymbol{(ii)}$ Fork, $Z \leftarrow X \rightarrow Y$; and $\boldsymbol{(iii)}$ Collider, $Z \rightarrow X \leftarrow Y$. If the causal structure of the variable combinations does not satisfy these local conditions according to the provided DAG $\mathcal{G}$, then the DAG is deemed invalid and is rejected.

D. Strategy Evaluation

In the final stage, the focus is on evaluating strategies for future actions within the causal framework, identifying causal flows. Actions of interest can be analyzed using counterfactuals, where treatment variables are examined in relation to parameters of interest to the management team in the domain of causal inference.

Counterfactuals are hypothetical scenarios that explore causal relationships and explain the effects of certain actions. Within causal graphical models, these scenarios bear a striking resemblance to the simulation of interventions. While the models focus on future outcomes, the analysis of counterfactuals directs attention toward an alternative historical context.

The central question addressed is whether there is a causal relationship between $X \rightarrow Y$. Given the interest in the potential outcomes of Y, it explores what would happen to Y if X had assumed the specific value $X = x$. This hypothesis is based on the counterfactual notation Y_x, which represents the hypothetical value of Y under the condition that $X = x$, providing a framework for evaluating the effects of such a change.

V. Results

The following section presents the results derived from the application of the method proposed in Section IV. This study was conducted in a payment processing company located in Chile, using a software factory as the case study, with the participation of seven individuals. The collected data covers a period of three months.

A. Event Log and Work Topology

The analyzed dataset comes from a record of events extracted from a platform oriented towards the traceability of requirements. This platform focuses on maintenance management with the goal of updating functionalities in accordance with the current legal regulations in the domain of payment methods. The record includes the participation of seven team members and comprises 480 entries with 15 variables, collected over a period of three months, covering 24 evolutionary projects. The data encompasses a variety of development projects focused on different platforms within the organization.

In the first phase, after applying the algorithm described in IV-B to identify the topology of interactions among the different roles contained in the event log, it is observed that the *"Functional Manager (Func Manager)"* and the *"Developer"*, highlighted in color, are essential components of this workflow. The Developer initiates the development process, which passes through the *"Tester"* until it reaches the *"Internal Client"*. Meanwhile, the *"Func Manager"* coordinates with the Developer and communicates with the *"Business Analyst"*, ensuring a constant flow of information and tasks that connect technical development with business needs. This structure allows for an understanding of the integration of daily activities without prior expert knowledge, showing how each role synchronizes with the requirements. This approach differs from previous works as it enables the identification of the structure without prior expert knowledge.

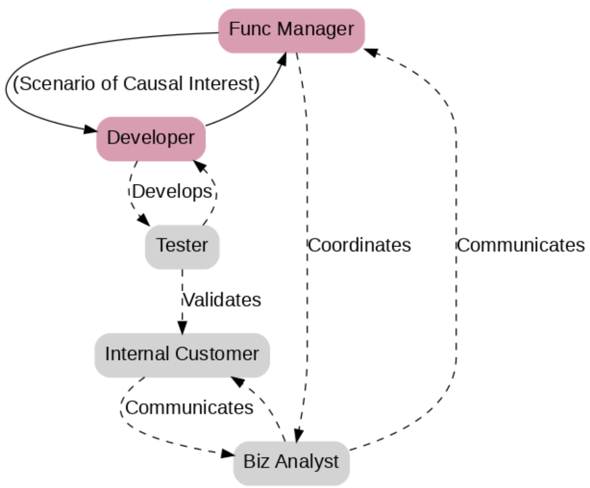

Fig. 4. Structure of relationships among the various roles contained in the event log from the execution of requirements.

For this phase of the analysis, four main variables were focused on: (i) the number of requirements (V_1), (ii) the time dedicated to unit testing (V_4), (iii) coordination time (V_3), and (iv) delivery delay time (V_2), as they are essential for evaluating the efficiency and effectiveness of the development process. The interaction between the Functional Manager and the Developer is crucial in this context, as the variables of coordination time and delivery delay time directly influence the efficiency and synchronization of the requirements. The organization is interested in the causal scenario between both roles due to the delivery delays of the requirements. Therefore, the goal is to identify the underlying causal scenario among the various recorded times. The objective is to interpret the causal direction between the variables and explain how focused actions can effectively address this problem.

B. Causal Graph

To identify and interpret the causes of delivery delays in requirements, functional causal models were used. These models, which assume an additive noise model, are based on the knowledge of the functional forms of the distributions that govern the relationships between variables, allowing for a precise identification of causal connections in the dataset. A representative algorithm of this category is DirectLiNGAM [31].

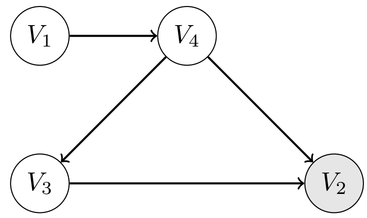

Fig. 5. Underlying causal structure in the dynamics of the variables present between the roles of *functional manager* and *developer*.

The discovered causal graph provides a clear view of the relationships between variables in the requirements delivery process. The number of requirements V_1 is the starting point, directly influencing the time spent on unit testing V_4. An increase in requirements V_1 extends the time needed for unit testing V_4 due to the higher number of test cases required. This increase in V4 has two effects: it contributes to delays in the delivery of requirements V_2 and affects coordination time V_3.

Coordination time V_3, influenced by V_4, also impacts delays in V_2. This indicates that more time spent on unit testing may require additional coordination to resolve issues during testing. The graph demonstrates how these variables, from the number of requirements V_1 to coordination time V_3 and unit testing V_4, lead to delays in requirements delivery V_2, revealing the complexity of the process and the multiple causal relationships that can cause delays in the final delivery.

The robustness of this analysis lies in its ability to capture causal information without the need for prior expert domain knowledge, unlike related works that assume relationships that are not necessarily causal but rather associations or correlations within the domain. This has allowed for the quick and precise identification of the relationships among key variables that require attention to assess potential improvements. The ability to autonomously identify these causal relationships facilitates more informed decision-making and the implementation of focused actions to optimize the requirement delivery process.

To refute the validity of the DAG, it was necessary to test each of the conditional independence constraints implicit in the graph. Since the relationships are linear causal, it is validated, for example, that $V_4 = f(V_1) + \epsilon$ and $V_1 \perp\!\!\!\perp \epsilon$.

The analysis of the causal graph reveals no significant violations of the Local Markov Conditions (LMC), with a p-value of 0.50. This suggests that the proposed graph breaks fewer "rules" than 50% of random graphs, indicating that it is likely

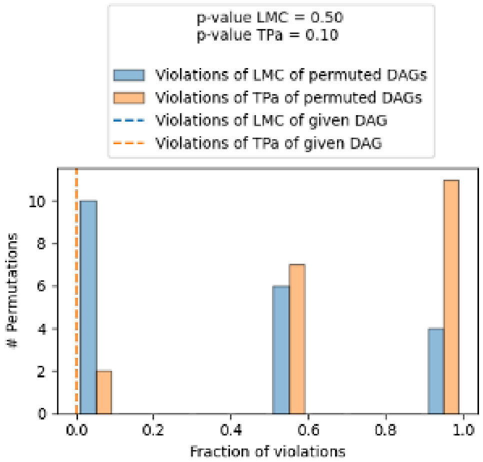

Fig. 6. Refutation of the validity of the causal graph in relation to the delay times in requirement delivery in the interaction between the Functional Manager and the Developer.

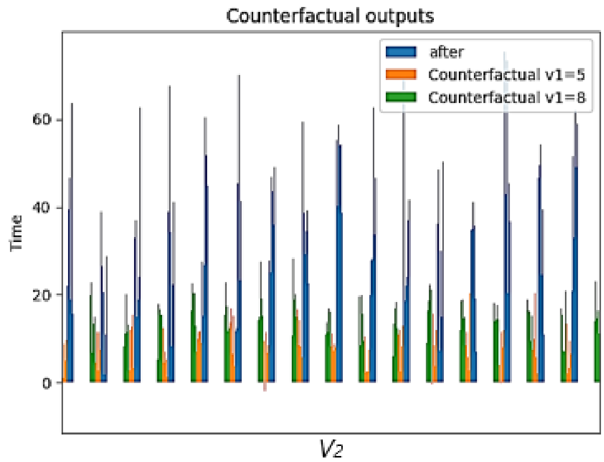

Fig. 7. In the counterfactual scenario conducted, the dataset is evaluated to determine what would have happened to V_2 if V_1 had been controlled in terms of the number of requirements, omitting their complexity.

a good reflection of the relationships in the data. Additionally, the p-value of 0.10 in the Triple Parentally D-Separated (TPa) test suggests that the graph is relatively informative and specific. This value indicates that only 10% of random graphs are as specific as the proposed one, meaning that the graph is quite unique and useful for understanding causal relationships. The distribution of violations shows that most random graphs have a fraction of LMC and TPa violations between 0.6 and 1.0. In contrast, the proposed graph has a fraction of violations close to 0 for both tests, indicating that it performs better than most random graphs. In summary, the proposed causal graph appears to be a reasonable and quite informative model of the causal relationships in the data. It does not significantly violate the LMC conditions and is more specific than most random graphs. Although it is not perfect, especially in terms of TPa, it provides a recommended representation of the underlying causal structure.

C. Counterfactual Analysis

Based on the delay times, two strategies were evaluated to improve the interaction between managers and developers: reducing the number of functional requirements per project V_1 to a range of 5 to 8. This counterfactual analysis allows for the exploration of the potential outcomes that would have been achieved by implementing this measure in the operational process.

VI. Conclusions and Future Work

The proposed method analyzes the causal relationships between roles in development teams, integrating organizational mining and causal inference. This enables early identification of team dynamics and the evaluation of targeted improvement implementations to enhance operational performance in the interaction and flow of activities between the functional manager and developer roles. The results obtained allow us to develop answers to the questions posed in I-C:

A. Q_1 : *"What is the underlying causal structure in the operational interactions among the team roles?"*

As a result of the interaction between the roles, the identification of key variables relevant to the analyzed scenario was facilitated. This process allowed the isolation of distributional noise, which in turn facilitated the discovery of causal graphs and the understanding of their functional relationships. This is essential for formulating effective questions about possible future scenarios and evaluating their implications.

B. Q_2 : Once the causal structure has been identified, *"which instances within the topological space are most likely to respond favorably to improvement interventions?"*

In response to the question, the developed counterfactual analyses have revealed which values are more viable for implementing improvements. For instance, we have found that reducing the number of requirements is causally linked to a decrease in the time spent on unit testing, which in turn directly impacts the reduction of delivery delays. In this context, opting to decrease the number of requirements to 8 presents a favorable and economically more viable alternative (lower demand) compared to reducing it to 5, while achieving a similar effect, as illustrated in Fig.7.

In this pilot test, only linear relationships were examined, excluding elements of the operational framework and complex causal relationships. It is crucial to investigate nonlinear scenarios, considering possible automation factors. Additionally, not all interactions in the process are significant due to weak connections, which could limit the effectiveness of the proposed method due to the insufficiency of records and operational variables.

The application of the proposed method in this work does not imply identifying a single causal graph for all variables in the context. Identifying one or more graphs over the same set of variables is because the instances of execution among the roles have different natures, such as the type of project for the type of client, among others.

Regarding future work, it is of interest to address operational relationship scenarios within business areas, specifically causal flow relationships. For example, it would be possible to explore both internal and outsourced development areas, using a comparative framework of causal relationships.

REFERENCES

[1] Ipek, Ozkaya. (2020). The Deployment View. IEEE Software.

[2] Danila, Moriakov., Ilsur, Shartdinov., Artem, Kruglov., Giancarlo, Succi. (2022). Team and Process Organization in Software Engineering.

[3] José, Adson, ...(2015). Towards a substantive theory of project decisions in software development project-based organizations: A cross-case analysis of IT organizations from Brazil and Portugal.

[4] Cielo, ...(2022). Uses of business process modeling in agile software development projects. Information Software Technology.

[5] (2022). Process Mining Model Integrated with Control Flow, Case, Organizational and Time Perspectives in a Software Development Project.

[6] Minseok, Song., Wil, M., P., van, der, Aalst. (2008). Towards comprehensive support for organizational mining.

[7] Amit, V., Deokar., Jie, Tao. (2021). OrgMiner: A Framework for Discovering User-Related Process Intelligence from Event Logs. Information Systems Frontiers.

[8] Singh, H., & Sharma, R. (2012). Role of adjacency matrix & adjacency list in graph theory. International Journal of Computers & Technology.

[9] Clark, Glymour., Kun, Zhang., Peter, Spirtes. (2019). Review of Causal Discovery Methods Based on Graphical Models.. Frontiers in Genetics.

[10] Alessio Zanga, Elif Ozkirimli, Fabio Stella, A Survey on Causal Discovery: Theory and Practice, International Journal of Approximate Reasoning.

[11] B. Schölkopf et al., ”Toward Causal Representation Learning,” in Proceedings of the IEEE, 2021.

[12] Janine, Witte.,... (2019). Covariate selection strategies for causal inference: Classification and comparison. Biometrical Journal.

[13] Digitale, J. C., Martin, J. N., Glymour, M. M. (2022). Tutorial on directed acyclic graphs. Journal of Clinical Epidemiology.

[14] Matthew J. Vowels, Necati Cihan Camgoz, and Richard Bowden. D'ya Like DAGs? A Survey on Structure Learning and Causal Discovery. ACM Comput. Surv. 2022

[15] Van der Aalst, W. M. P., Adriansyah, A., Alves de Medeiros, A. K., Arcieri, F., Baier, T., Blickle, T., ... & Wynn, M. T. (2011). Process Mining Manifesto.

[16] Liuyi, Yao., Zhixuan, Chu., Sheng, Li., Yaliang, Li., Jing, Gao., Aidong, Zhang. (2020). A Survey on Causal Inference. arXiv: Methodology,

[17] Yao, L., Chu, Z., Li, S., Li, Y., Gao, J., & Zhang, A. (2021). A survey on causal inference. ACM Transactions on Knowledge Discovery from Data (TKDD).

[18] H. Jiang, P. Qi, ..., ”A Short Survey on Forest Based Heterogeneous Treatment Effect Estimation Methods: Meta-learners and Specific Models”, Conferencia Internacional IEEE 2021 sobre Big Data (Big Data).

[19] Devriendt, F., Moldovan, D., & Verbeke, W. (2018). A literature survey and experimental evaluation of the state-of-the-art in uplift modeling: A stepping stone toward the development of prescriptive analytics. Big data.

[20] Holland, P. W. (1986). Statistics and causal inference. Journal of the American Statistical Association.

[21] Gutierrez, P., & Gérardy, J. Y. (2017, July). Causal inference and uplift modelling: A review of the literature. In International conference on predictive applications and APIs.

[22] Pierluigi Zerbino, Alessandro Stefanini, Davide Aloini,Process Science in Action: A Literature Review on Process Mining in Business Management,Technological Forecasting and Social Change.

[23] Elias, Eulig., Atalanti, ... (2023). Toward Falsifying Causal Graphs Using a Permutation-Based Test.

[24] Haneen, Hijazi., Shihadeh, Alqrainy., Hasan, Muaidi., Thair, Khdour. (2013). Identifying Causality Relation between Software Projects Risk Factors.

[25] Fausto, Guaragni., Kristin, Paetzold., Roland, Ortt. (2018). Understanding the Cause-Effects Relations of Severe Influences Affecting Ongoing Product Development Projects

[26] Nicolli, Rios., Rodrigo, O., Spínola., Manoel, Mendonça., Carolyn, Seaman. (2019). Supporting analysis of technical debt causes and effects with cross-company probabilistic cause-effect diagrams.

[27] Woo, Sung, Jang., Roger, Kim. (2022). Automatic Cause–Effect Graph Tool with Informal Korean Requirement Specifications. Applied Sciences.

[28] Prateeti, Mohapatra., Petra, Björndal., Karen, J., Smiley. (2010). Causal Analysis of Factors Governing Collaboration in Global Software Development Teams.

[29] Laura-Diana, Radu. (2019). Effort Prediction in Agile Software Development with Bayesian Networks.

[30] Arthur, Silva, Freire., Mirko, Perkusich., Renata, M., Saraiva., Hyggo, Almeida., Angelo, Perkusich. (2018). A Bayesian networks-based approach to assess and improve the teamwork quality of agile teams. Information & Software Technology.

[31] Shimizu, S., Inazumi, T., Sogawa, Y., Hyvarinen, A., Kawahara, Y., Washio, T., ... & Hoyer, P. . DirectLiNGAM: A direct method for learning a linear non-Gaussian structural equation model. Journal of Machine Learning Research-JMLR, 12(Apr), (2011).

4.4 Causal Learning: Monitoring Business Processes Based on Causal Structures

Building on the previous structural and organizational results, the fourth article places the causal model in the day-to-day operation of the process. **CaProM** integrates the learned causal structure into the interventional monitoring of temporal performance ($\mathbf{L}_2$). In this way, deviations are analyzed as possible changes in the causal mechanisms and not only as statistical anomalies (**SO.2**, **RQ.2**). Thus, the overall contribution shifts from model formulation to its systematic use in monitoring. At the domain level, the work broadens the process monitoring literature by incorporating a causal reading of temporal deviations.

Publication Summary

Title:	*Causal Learning: Monitoring Business Processes Based on Causal Structures*
Authors:	Fernando Montoya, Hernán Astudillo, Daniela Díaz and Esteban Berríos
Special Issue:	Causal Graphical Models and Their Applications
Journal:	*Entropy*
Year:	2024
Doi:	https://doi.org/10.3390/e26100867

entropy

Article

Causal Learning: Monitoring Business Processes Based on Causal Structures

Fernando Montoya [1,2,3,*], Hernán Astudillo [4,*], Daniela Díaz [5] and Esteban Berríos [3]

1 Nexus Payment Systems SpA, Santiago 8320123, Chile
2 Campus Casa Central, Universidad Técnica Federico Santa María, Valparaíso 2390123, Chile
3 Fundación Instituto Profesional Duoc UC, Santiago 8240000, Chile; es.berriosp@duocuc.cl
4 Instituto de Tecnología para la Innovación en Salud y Bienestar, Universidad Andres Bello, Viña del Mar 2530958, Chile
5 IT Solution SpA, Santiago 8320000, Chile; ddiazb4@correo.uss.cl
* Correspondence: fernando.montoya@usm.cl (F.M.); hernan@acm.org (H.A.)

Abstract: Conventional methods for process monitoring often fail to capture the causal relationships that drive outcomes, making hard to distinguish causal anomalies from mere correlations in activity flows. Hence, there is a need for approaches that allow causal interpretation of atypical scenarios (anomalies), allowing to identify the influence of operational variables on these anomalies. This article introduces (*CaProM*), an innovative technique based on causality techniques, applied during the planning phase in business process environments. The technique combines two causal perspectives: *anomaly attribution* and *distribution change attribution*. It has three stages: (1) process events are collected and recorded, identifying flow instances; (2) causal learning of process activities, building a directed acyclic graphs (DAGs) represent dependencies among variables; and (3) use of DAGs to monitor the process, detecting anomalies and critical nodes. The technique was validated with a industry dataset from the banking sector, comprising 562 activity flow plans. The study monitored causal structures during the planning and execution stages, and allowed to identify the main factor behind a major deviation from planned values. This work contributes to business process monitoring by introducing a causal approach that enhances both the *interpretability* and *explainability of anomalies*. The technique allows to understand which specific variables have caused an atypical scenario, providing a clear view of the causal relationships within processes and ensuring greater accuracy in decision-making. This causal analysis employs cross-sectional data, avoiding the need to average multiple time instances and reducing potential biases, and unlike time series methods, it preserves the relationships among variables.

Keywords: causal graph; causal attribution of anomalies; causal attribution of distributional change; business process monitoring; business process mining

check for updates

Citation: Montoya, F.; Astudillo, H.; Díaz, D.; Berríos, E. Causal Learning: Monitoring Business Processes Based on Causal Structures. *Entropy* **2024**, *26*, 867. https://doi.org/10.3390/e26100867

Academic Editors: Luis Enrique Sucar and David Danks

Received: 18 August 2024
Revised: 1 October 2024
Accepted: 8 October 2024
Published: 15 October 2024

1. Introduction

The monitoring of business processes has become a critical challenge in modern management due to the dynamic environment, increasing complexity, and interconnection of activity flows [1]. These factors demand methods that go beyond simple metric evaluations. Traditionally, such oversight has relied on static models, predefined metrics, and comparisons of established execution patterns, which limit the ability to identify and understand the true underlying causal relationships in business processes. In this context, there is a need for an approach that facilitates not only the observation but also the interpretation and explanation of business process behavior from a causal perspective.

Despite advances in predictive [2–4], as well as prescriptive [5,6], analysis techniques for process monitoring, such techniques are still primarily based on patterns of correlation or association, without delving into causal reasoning. This limits the ability to identify the true underlying structures that govern operations. While correlation can indicate relationships between variables, it does not necessarily establish a cause–effect relationship. Without a methodology that explains the implications of causal effects, the understanding of operational dynamics remains incomplete, which can lead to biased decisions and hinder effective problem solving. This underscores the need for a causal approach that identifies causes and interprets their implications, improving decision making in complex business environments. Throughout a business process flow, multiple distinct instances may arise due to the diversity of operational objectives. This results in varied and often complex causal relationships. Therefore, it is essential to implement advanced approaches that go beyond traditional monitoring. These approaches must be capable of interpreting and thoroughly explaining the variations in business processes, grounded in a solid causal structure.

The remainder of this paper is structured as follows: Section 2 formalizes the problem; Section 3 surveys related work; Section 4 explains the proposed technique; Section 5 presents the case study; Section 6 discusses the results; and Section 7 summarizes and concludes.

2. Context and Problem

This section formalizes the description of the research problem as one of causal reasoning on business processes.

2.1. Problem Description

Figure 1 illustrates a business process flow in which execution instances are represented as tokens corresponding to specific tasks defined in the planning of various projects. In this context, organizational areas continuously monitor a series of key performance indicators (KPIs) to manage and optimize the processes. These KPIs include compliance with service-level agreements (SLAs); resource utilization; and other critical factors, such as hours allocated to specific tasks, personnel assigned to essential roles, and the achievement of departmental objectives [7]. This systematic approach enables the measurement of anomalies in the process flow, based on the defined metrics, which manifest in critical scenarios such as those described in *A* and *B*.

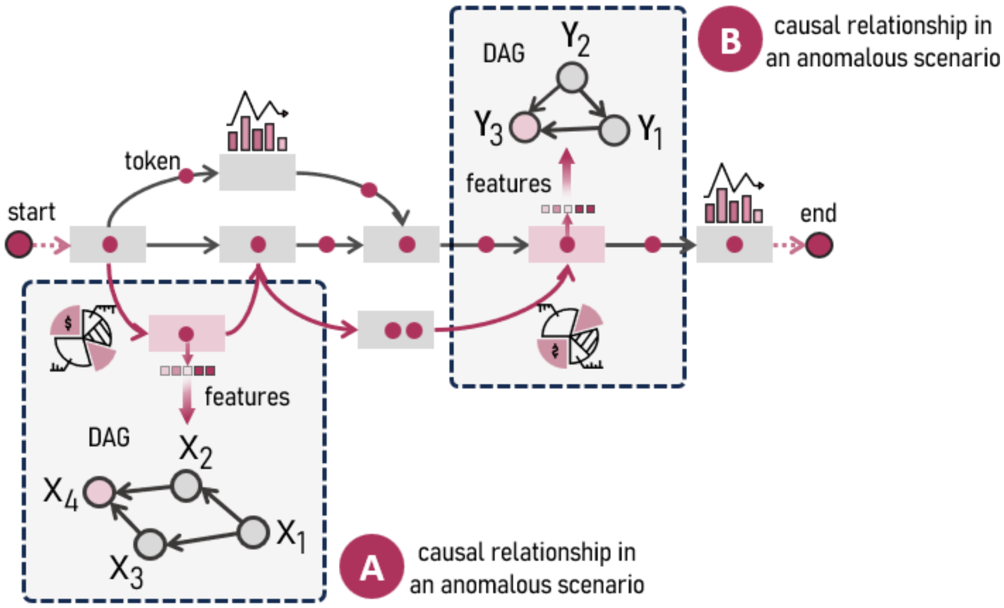

Figure 1. Characterization of causal variables in business process activities during anomalous scenarios with respect to the effect variables exhibiting anomalies.

Interpreting and explaining the causal relationships in these scenarios presents a considerable challenge, as it involves uncovering the underlying causal relationships connecting the various events and metrics involved. Moreover, understanding the causal direction of the variables that trigger these events is crucial to ensuring more precise and proactive process management, allowing for timely adjustments in resource allocation and task planning.

Approaching this analysis from a causal perspective establishes the directed acyclic graph (DAG) as a key tool for understanding the connections between variables [8,9], as DAGs not only aid in identifying the causes of anomalous events but also provide a more comprehensive understanding of the operational interactions within the process. Nevertheless, their implementation demands a deep understanding of causal topology. In Figure 1, the DAGs from scenarios A and B reveal different levels of complexity, where, within the same process, the operational aspects of two distinct activities are represented by the variables X_i and Y_i, respectively. Both significantly influence the overall flow, as their interactions can determine both the stability and performance of the process, potentially leading to anomalous scenarios in the business process flow.

2.2. Causal Reasoning in the Business Process Context

Business process monitoring not only entails supervising key indicators but also examining how the relationships between variables influence overall performance. Understanding these relationships from a causal perspective is crucial for improving decision making and optimizing operational performance. Through causal learning, it is possible to identify how variables interact and impact the process flow, enabling early detection and correction of anomalies. Scenarios A and B (see Figure 1) illustrate how this causal approach can influence different activities within the same business process, given the distinct causal topologies between variables X_i and Y_i.

In Scenario A, variables X_1, X_2, X_3, and X_4 are part of a specific activity within the process, where X_1 directly influences X_2 and X_3 and these two, in turn, affect X_4. Applying causal learning shows that X_2 and X_3 are independent of each other when conditioned on X_1, which is expressed as $X_2 \perp\!\!\!\perp X_3 \mid X_1$; however, when conditioned on X_4, which acts as a collider, X_2 and X_3 become dependent. This understanding introduces important considerations for business process decision making because if an anomaly occurs in X_4, causal learning identifies that the potential causes lie in X_2 and X_3. Due to the conditional independence between these variables given X_1, it is possible to analyze each one separately to determine which is contributing to the anomaly, facilitating more effective mitigation by allowing interventions to be specifically directed at the problematic variable without unnecessarily affecting other parts of the process.

On the other hand, in scenario B, corresponding to a different activity within the same process flow, the variables Y_1, Y_2, and Y_3 interact differently. Y_2 directly influences both Y_1 and Y_3, and Y_1 also influences Y_3. This creates a causal chain where Y_2 affects Y_3 both directly and indirectly through Y_1, indicating that all influences must be considered when analyzing Y_3. Causal learning indicates that an anomaly in Y_3 could be the result of multiple inter-related factors, requiring decision makers to account for both the direct and indirect influences of Y_2 and Y_1. Mitigating the anomaly, therefore, requires a comprehensive approach addressing all the variables involved to identify and resolve the root cause of the problem.

These scenarios highlight the importance of having previous causal insights and mechanisms for interpreting anomalies using DAGs. Without a thorough understanding of the causal relationships and dependency topology in business processes, decision making can become inefficient. Implementing causal learning and developing tools that interpret anomalies according to the causal structure allow for the identification of underlying causes and the design of effective interventions. This approach not only facilitates the resolution of current anomalies but also helps to prevent future incidents, ultimately improving the overall performance of the business.

2.3. Research Questions

This work proposes a monitoring methodology based on two causal perspectives, anomaly attribution and distributional change attribution [10,11]. This approach allows for identifying the contributions of each node to the observed anomalies and changes in the system's causal mechanisms, facilitating a deep understanding of the variations and their underlying causes.

2.3.1. Anomaly Attribution

The purpose of these questions is to evaluate how the causal connections between variables explain the severity of observed anomalies [10].

- RQ_1: *What is the causal contribution of a specific variable (e.g., X_1 or Y_2) to the anomaly observed in a target variable (e.g., X_4 or Y_3), taking into account the entire causal graph?* The influence of variables like X_1 or Y_2 is key to understanding the origin of anomalous deviations in the system, allowing for the identification of critical points that generate these unexpected behaviors.
- RQ_2: *How do causal paths, such as $X_1 \rightarrow X_2 \rightarrow X_4$ or $Y_2 \rightarrow Y_1 \rightarrow Y_3$, affect the severity of the anomalies observed in X_4 and Y_3?* Causal paths, such as $X_1 \rightarrow X_2 \rightarrow X_4$ or $Y_2 \rightarrow Y_1 \rightarrow Y_3$, break down the internal dynamics of the system, revealing how intermediate interactions contribute to the final magnitude of the observed anomaly.

2.3.2. Distributional Change Attribution

The focus of these questions is to analyze how changes in causal mechanisms and interactions between nodes affect the observed distributions in different scenarios [11,12].

- RQ_3: When comparing two different contexts, *how can we identify and quantify the changes in the distributions of target variables (e.g., X_4 or Y_3), and what evidence suggests that these differences are due to alterations in the causal mechanisms of specific variables (e.g., X_1 or Y_2)?* This question aims to identify which nodes experience changes in their mechanisms and how they impact the distribution of the target variables under different conditions.
- RQ_4: *How do changes in causal interactions within paths $X_1 \rightarrow X_2 \rightarrow X_4$ and $Y_2 \rightarrow Y_1 \rightarrow Y_3$ impact the distributional properties of X_4 and Y_3?* This question examines how modifications in the causal relationships within specific paths alter the distributions of variables X_4 and Y_3.

3. Related Work

Recent literature on business process monitoring has explored causal approaches to improving efficiency and effectiveness in management. Traditionally, causal inference for estimation of effects has largely depended on expert knowledge, which may introduce potential limitations and biases in the analysis. However, there is a growing need to apply causal learning techniques to enable the autonomous discovery of causal relationships.

For example, Shoush et al. [13] used algorithms such as orthogonal random forests (**ORF**) to optimize resource allocation and intervention policies, leveraging historical data to predict needs in credit origination. This approach is complemented by the work described by Bozorgi et al. [14], which combines causal inference with reinforcement learning to enhance treatment policies, maximizing net benefits in business processes. Both studies emphasize the importance of accurately evaluating the effects of interventions, although they depend on expert knowledge to define initial causal relationships.

In a similar context, Mehdiyev et al. [15] addressed uncertainty in predictive process monitoring by integrating information systems, machine learnig, and operations research techniques. By using quantile regression forests (**QRF**) and Shapley Additive Explanations (**SHAPs**), they provided a deeper understanding of model uncertainties. This methodology resonates with the findings of Wang et al. [16], who applied deep learning techniques such as **CNN** and **BigRu** to capture dependencies in business processes, improving the prediction of future activities. Both approaches underscore the importance of managing uncertainty and dependencies in data to enhance intervention decisions.

Furthermore, Kotsias et al. [17] integrated deep learning techniques into Business Process Management (**BPM**) through process mining. Their innovative approach combined predictive and prescriptive monitoring using Reinforcement Learning (**RL**) in the banking sector. Methods such as LSTM and Q-learning were employed to predict future states and optimize decisions, highlighting the Inductive Miner, a process discovery algorithm, for its flexibility and scalability. Results showed that the RL model outperformed traditional methods, suggesting future research in RL algorithms and deep learning for complex business environments.

Zahra et al. [18] introduced a prescriptive approach that applies causal inference to estimate the effects of interventions in real time, particularly in the execution of approval request processes. This method adjusts intervention decisions based on user-defined policies and is related to work aiming to improve accuracy in measuring causal effects, as detailed by Pavlos Delias et al. [19]. In this context, doubly robust estimation techniques were employed to mitigate biases and enhance the precision of effect estimates, using simulated process execution data.

Building on the theme of improving predictive accuracy in business processes, Jens Brunk et al. [20] focused on predicting future events by considering context, using dynamic Bayesian networks to model cause–effect relationships. This approach, which combines predictive analytics and machine learning, aligns with the efforts of the previously mentioned articles in seeking to improve the accuracy of predictive process monitoring (**PPM**). However, as in other studies, causality is often assumed based on expert knowledge, which can limit the generalization of the models.

Focusing on the analysis of time series in business processes, Hompes et al. [21] present a technique for identifying causal factors that influence process performance. Their approach, based on Granger causality and combining domain knowledge with advanced clustering and filtering techniques, enables the systematic analysis of event logs to generate causal graphs that reveal underlying relationships between different process variables. The proposed technique includes the creation of a decomposition graph and an inclusion graph, facilitating the discovery of complex interactions between process features and performance metrics.

In summary, there is a need to apply independent and objective causal discovery techniques for anomaly detection in business process monitoring. This approach aims to reduce reliance on expert knowledge and enable a comprehensive interpretation of anomalous event causality, facilitating effective and targeted interventions.

4. Proposed Technique

This article presents *CaProM*, an innovative technique for business process monitoring, focused on managing anomalies caused by noise in variables and shifts in the distribution of causal graphs. Unlike previous approaches that rely on expert knowledge [13,14], *CaProM* employs automated causal discovery [22,23], setting itself apart from time series based approaches [21]. It involves independently monitoring causal graphs within each activity of the process flow, providing a deeper understanding that goes beyond effect estimation by identifying the variables that significantly contribute to anomalous events. *CaProM* facilitates the determination of specific actions to address concrete events within the process.

CaProM has three stages. First, process events are collected and recorded from project plans, reflecting flow instances that will be executed in the business process. Second, causal learning is carried out by constructing independent directed acyclic graphs (**DAGs**) for each process activity, which represent the dependencies between variables. Third and finally, these DAGs are used to monitor the process, detecting anomalies and critical nodes, based on techniques [10,11].

4.1. Project Event Log

CaProM's first stage (Figure 2A) applies process mining techniques to event logs from several projects, to capture the complete model that will be executed. This provides a more accurate representation of operational reality, overcoming the limitations of documented process flows that are often not strictly followed. Event logs for the projects can be sourced from Gantt schedules, and estimates of operational variable usage can be provided by several process roles. Analyzing aggregated traces from the projects allows to identify critical activities and process variants.

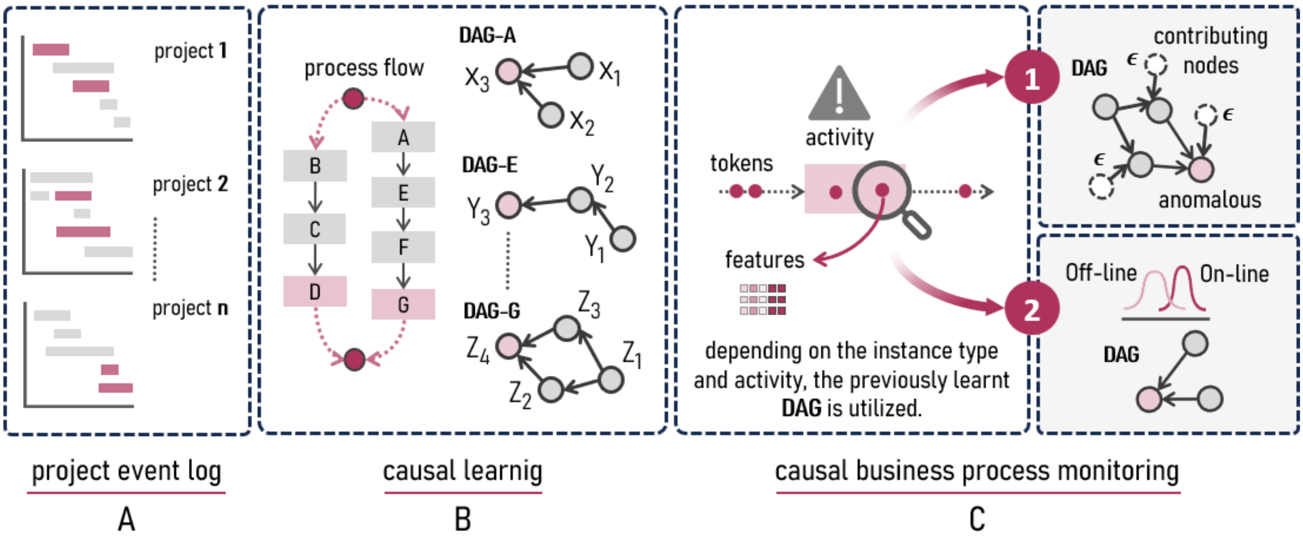

Figure 2. Stages of the proposed *CaProM* technique: event logging, causal learning, and monitoring of anomalies.

For example, software development projects have critical activities proper to software development (e.g. integration and testing), and changes in their requirements impact their execution effort. This first stage is essential to understand the actual execution of processes and uncover patterns and deviations that are not reflected in the official documentation.

In addition, the event log in project planning, as shown in Table 1, includes elements such as the event identifier linked to the project, the planned activity, the timestamp showing the allocated time for completing the activity, the responsible person, and the associated operational variables. These logs detail each stage of the project, allowing for precise tracking of the actual planning for its execution.

Table 1. Project event log.

Project ID	Activity	Time Span	Role$_1$	Feature$_1$	Feature$_2$	…	Feature$_n$
proj.$_1$	A	350.2 ms	role.$_1$	-	feat.$_2$	…	feat.$_n$
proj.$_1$	B	270.2 ms	role.$_2$	feat.$_1$	feat.$_2$	…	-
proj.$_1$	C	450.5 ms	role.$_4$	-	feat.$_2$	…	feat.$_n$
proj.$_2$	A	510.4 ms	role.$_5$	feat.$_1$	-	…	feat.$_n$
proj.$_2$	B	350.8 ms	role.$_1$	feat.$_1$	feat.$_2$	…	-
…	…	…	…	…	…	…	…
proj.$_n$	A	230.4 ms	role.$_5$	-	feat.$_2$	…	feat.$_n$
proj.$_n$	B	130.6 ms	role.$_2$	feat.$_1$	-	…	feat.$_n$
proj.$_n$	C	310.2 ms	-	feat.$_1$	feat.$_2$	…	-

4.2. Causal Learning

CaProM's second stage (Figure 2B), focuses on the causal analysis of the activities in the process model, obtained from the event logs of multiple projects. Each activity considers a set of operational variables $V^a = \{X_1^a, \ldots, X_n^a, Y_1^a, \ldots, Y_m^a\}$. Using causal discovery algorithms specifically designed for cross-sectional data, DAGs are constructed to represent the causal relationships between these variables, based on simultaneous observations of multiple instances of the activity.

The main goal of this technique is identifying the causal graph that correctly describes the interactions among operational variables, to allowing understanding the causal topology and the specific interrelationships within each process activity. To this end, it employs an algorithm based on the Linear Non-Gaussian Acyclic Model (LiNGAM) that assumes linear relationships and non-Gaussian error terms, thus enabling identification of causal direction even in complex situations with cross-sectional data [24].

$$Y_i^a = f_a(\theta_a, X_i^a) + \epsilon_{Y_i}^a \tag{1}$$

where Y_i^a represents the effect variable of activity a_i for process instance i; X_i^a is the specific causal variable; θ_a denotes the global parameters; and $\epsilon_{Y_i}^a$ is the non-Gaussian error term. In addition to LiNGAM, other causal discovery algorithms from the following two main families are employed: *constraint-based* and *score-based methods*.

Constraint-based algorithms, such as the Peter–Clark algorithm (PC), determine the causal structure by conducting conditional independence tests, systematically evaluating whether $X_1^a \perp\!\!\!\perp Y_1^a \mid Z$ for variables X_1^a, Y_1^a, and sets of variables Z, thereby determining the existence or absence of edges in the dependency graph (G_{PC}). PC starts with a fully connected graph and removes edges based on detected conditional independencies, resulting in a Partially Directed Acyclic Graph (CPDAG).

Score-based algorithms, such as Greedy Equivalence Search (GES), search for the structure that maximizes a scoring function ($S(G)$). GES operates by performing a greedy search in the space of causal structures, finding $G^* = \arg\max_G S(G)$, where G^* is the optimal graph according to the scoring function.

Figure 3 illustrates the generalized causal model derived from the causal discovery algorithms [22,23]. This representation explicitly captures the global parameters (θ_a) associated with the effect variables (Y_i^a), as well as the error terms ($\epsilon_{X_i}^a$ and $\epsilon_{Y_i}^a$), allowing LiNGAM to exploit non-Gaussianity to infer causal direction. In the same graph, the PC and GES algorithms integrate θ_a into their methodologies; PC implicitly utilizes θ_a in the conditional independence tests to determine the dependency relationships between the effect variables (Y_i^a), while GES optimizes a scoring function that incorporates θ_a to evaluate and select the most suitable causal structure from the space of possible graphs.

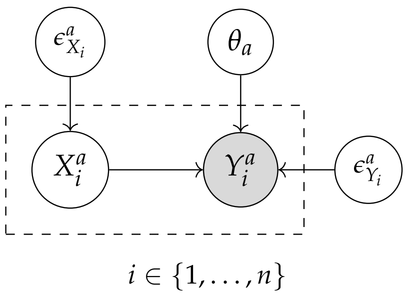

Figure 3. Generalized causal graph illustrating the causal relationships between operational variables, where X_1^a has a causal effect on Y_1^a, situated in activity a of the business process.

The causal model for the variables involved in each business process activity can be expressed in terms of their joint probability as follows:

$$P(V^a, \theta_a) = P(\theta_a) \prod_{i=1}^{n} P(Y_i^a | X_i^a, \theta_a) \tag{2}$$

where this probabilistic framework facilitates the integration of various causal discovery algorithms to infer the causal structure and estimate the functional relationships between the operational variables. The factorization of the joint probability reflects the structure of the DAG obtained by the algorithms, with each variable (Y_i^a) depending on its direct causes

(X_i^a) and the parameters (θ_a). Variables (X_i^a) can be considered exogenous or dependent on other variables within the causal model.

4.3. Causal Monitoring

In *CaProM*'s third stage, process flow monitoring is based on the previously established DAG, to identify and examine anomalous events in the operational variables. Figure 2C illustrates the process of comparing the variables of interest with a predetermined threshold. When a variable exceeds the established operational limits, the analysis focuses on the causal variables that contributed to the anomaly. To address this causal analysis, two specific techniques are proposed, as described in the following subsections.

4.3.1. Causal Analysis of Anomalous Attributes

A causal analysis of anomalous attributes is conducted by identifying and understanding causal relationships among sources of atypical behavior, by evaluating the data captured during process planning [10], as documented in project event logs (Section 4.1). Additionally, this technique analyzes how noise present in key variables impacts the metrics of interest, to detect unusual patterns and attribute them to specific causes.

Based on the causal graph topology shown in Figure 4, the perturbations in the error terms ($\epsilon^a_{B_i}$, $\epsilon^a_{C_i}$, and $\epsilon^a_{Y_i}$) are evaluated, and their effects on Y_i^a are analyzed. Initially, an outlier score for Y_i^a is established using anomaly detection models such as the difference of means. Subsequently, the contributions of B_i^a, C_i^a, and other causal nodes are quantified, determining how variations in $\epsilon^a_{B_i}$, $\epsilon^a_{C_i}$, and $\epsilon^a_{Y_i}$ influence the probability of Y_i^a exhibiting anomalous values. This impact is calculated as the probability that Y_i^a exceeds a threshold ($g(Y_i^a)$) when the values of $\epsilon^a_{B_i}$, $\epsilon^a_{C_i}$, and $\epsilon^a_{Y_i}$ are replaced with random values. it is measured as $-\log P(Y_i^a \geq g(Y_i^a) \mid \text{replace } \epsilon_i^a \text{ with random value}$, where the threshold $g(Y_i^a)$ (e.g. difference between means) determines whether Y_i^a is anomalous. This logarithmic measure provides a clear assessment of the direct impact of the perturbations.

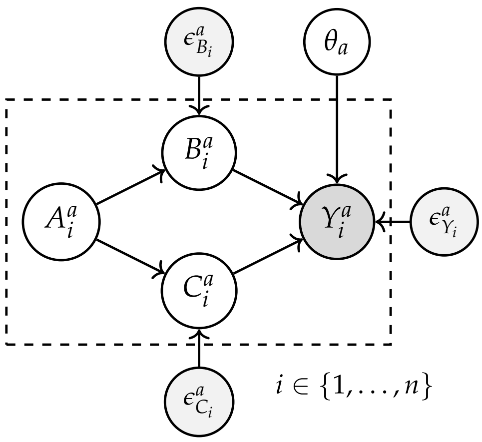

Figure 4. Characterization of the impact of noise on the variables of Y_i^a.

4.3.2. Causal Attribution of Distributional Changes

The causal attribution of distributional changes aims to identify how changes in variables, such as $X_i^a = \{A_i^a, B_i^a, C_i^a\}$ affect the distribution of the variable of interest (Y_i^a) [11,12]. In Figure 5, the nodes on the left represent the prior data ($P(Y_i^a \mid X_i^a)$) obtained from the project event logs, while the nodes on the right show the current data ($\hat{P}(Y_i^{a'} \mid X_i^{a'})$) collected during the current process execution of the process (see Equation (2)). This is crucial for business process monitoring, as it allows for an understanding of how changes in operational conditions affect key outcomes.

Unlike previous proposals [13,14,18,20], which focused on estimating causal effects and specific interventions, *CaProM* analyzes variations in the variables distribution. It thus provides a more comprehensive view of how operational environment changes influence the process, capturing not only direct causes but also noise and influences from previous activities. Moreover, the approach is independent of the intervention training, allowing a wider evaluation of operational conditions that could trigger anomalous events.

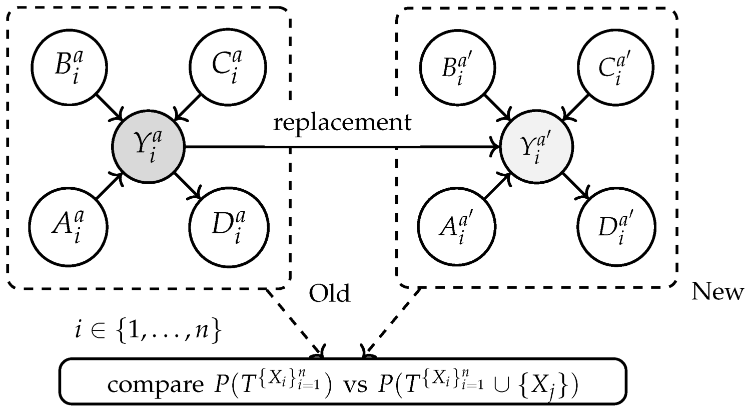

Figure 5. Characterization of distribution change attribution in a DAG for activity a_i.

The process begins with estimation of the conditional distributions of the variables using $P(Y_i^a \mid X_i^a)$, where X_i^a represents the set of direct causal variables that influence Y_i^a; These distributions reflect the planning and expectations based on previous events (project). Next, the previous causal mechanisms ($P(Y_i^a \mid X_i^a)$) are systematically replaced with the mechanisms based on the new data ($\hat{P}(Y_i^{a'} \mid X_i^{a'})$), which reflect the current reality of process execution. This replacement generates new marginal distributions for $Y_i^{a'}$, denoted as $\hat{P}(T^{X_n})$, allowing to observe how these changes affect the variable of interest.

The next step involves comparing the marginal distributions before and after the replacement, specifically $P(T^{\{X_i\}_{i=1}^n})$ versus $P(T^{\{X_i\}_{i=1}^n} \cup \{X_j\})$. Here, T represents the set of nodes considered in the analysis, and $\{X_i\}_{i=1}^n$ represents the variables of interest whose distributions are being measured. The notation $\cup\{X_j\}$ indicates the inclusion of the distribution of X_j with a set of distributions of X_i, where X_j is a new variable or a modification of an existing variable, being evaluated to see how its inclusion or change affects the distribution of the variables of interest.

This analysis helps to identify which node changes (e.g. A_i^a, B_i^a or C_i^a) are responsible for the observed variations. Understanding the changes in the conditional distributions ($\hat{P}(Y_i^{a'} \mid X_i^{a'})$) affect $Y_i^{a'}$ allows to identify the underlying causes of variations in the performance of process activities. In turn, this enables precise adjustments in operational variables to mitigate issues and optimize system performance.

5. Results

This section presents the results obtained by applying the *CaProM* technique described in Section 4, which adapts causal discovery techniques to enable causal learning in the planning and execution stages of a business process. The findings show the efficacy of *CaProM* in detecting and causally interpreting anomalies and distributional shifts during the monitoring of the execution stage. *CaProM* provides insights that go beyond traditional correlations in the analysis of business processes.

5.1. Dataset

The validation dataset includes information from 562 projects, each with specific technological and regulatory requirements for a group of banking entities. This dataset provides a comprehensive view of the business process, allowing for the monitoring of planned activities throughout the workflow, observing them in both the planning and execution phases (see Table 2). The stages share the same activities, although they are given different names depending on the phase. For example, an activity labeled *Preliminary Evaluation* during the planning phase may correspond to *Requirement Implementation* during the execution phase, both with the same objective of specifying and fulfilling project requirements.

Table 2. Process flow activities.

No.	Planning Activities	Execution Activities
1	Requirements Analysis	Integrity Review
2	Preliminary Evaluation	Requirements Validation
3	Functional Specification	Functional Implementation
4	Technical Design	Development and Integration
5	Test Design	Test Execution
6	Architecture Design	Architecture Validation
7	Project Approval	Deliverables Approval
8	Implementation Planning	Production Deployment
9	Request Closure	Project Closure

Additionally, the data follow the structure described in Section 4.1 based on event logging. Key elements include the *Instance ID* to identify each project, the *Activity Name* specifying the actions, the *Execution Times* capturing the duration, the *Roles* assigning responsibilities, and several operational variables.

5.2. From Project Plans to Process Map

The application of *CaProM* (see Figure 2B) allowed to discover the process model from the dataset derived from project planning (Section 4.1). The flow was generated using Apromore, a cloud-based process mining tool that employs an advanced discovery algorithm based on Split Miner [25]. The analysis identified the activities and their interactions, which are represented in Figure 6.

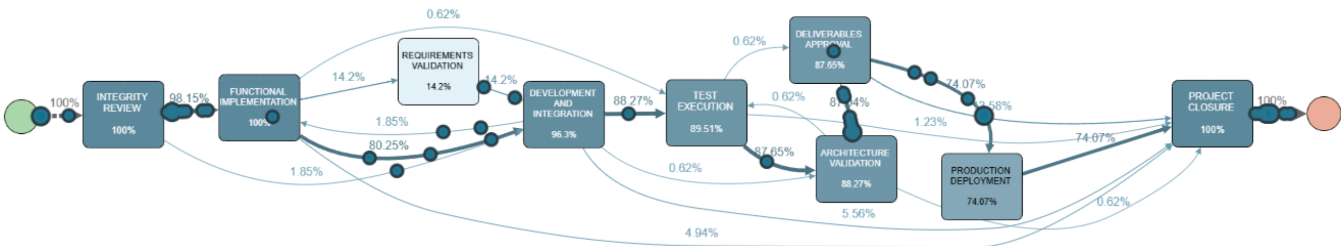

Figure 6. Process flow extracted from the project event logs using Apromore.

The resulting model provided an accurate visualization of the planning process structure. An activity that captures particular attention from managers is *Test Execution* due to the complexity of project requirements and variability in required hours. This phase, referred to as *Test Design* in project plans, includes testing tasks outsourced by the organization. The variation in the duration of these externalized tasks has a significant impact on other variables of the activity. This finding prompted the identification of causal relationships among variables in order to interpret the effect of outsourced tasks on the activity.

5.3. Operational Variables Extracted from Test Design for Causal Monitoring in Test Execution

Table 3 presents the key operational variables monitored during the *Test Execution* activity. These variables encompass various aspects of the process, such as code coverage, number of meetings, people involved, and test cases. Note that hour-related variables, such as *testing_hours* and *regression_time*, represent estimates of required quantities, not actual measurements over time. These estimates are made by those responsible during the planning phase, and are monitored throughout the test execution.

Table 3. Variables in the *Test Execution* activity.

No.	Variable	Description
1	test_coverage	Percentage of code or requirements covered by tests
2	num_meetings	Hours dedicated to meetings for coordination, clarification, and scope changes
3	testing_hours	Hours dedicated to testing
4	num_testers	Number of people involved in testing
5	regression_time	Time spent on regression testing after changes
6	complexity_level	Complexity level of the requirement
7	num_test_cases	Number of cases in the test plan
8	defects_found	Number of defects identified during testing
9	automation_level	Percentage of test cases that are automated
10	num_requirements	Number of associated functional requirements

The analysis confirmed the cross-sectional nature of the data, where each observation represents a unique instance of the process. Although the variables are recorded with a *Time Span* interval based on the project log (see Table 1), multiple execution instances exist at the same temporal point. Addressing temporality under these conditions would require the averaging of data from different simultaneously executed instances, i.e., calculation of an average value ($\bar{X}_t^a = \frac{1}{n_t}\sum_{i=1}^{n_t} X_{it}^a$) for each instant ($t$), where n_t is the number of instances at time t in activity a. However, this procedure can introduce biases and the loss of crucial information, affecting the estimation of the causal effect ($P(Y_i \mid do(X_i))$). Here, the notation $do(X_i)$ represents an intervention on the variable X_i, that is, evaluation an improvement action. Averaging could mask individual variations and temporal dynamics essential for causal discovery, hindering the identification of reliable structures.

Additionally, there is a risk of incurring in Simpson's paradox, where trends present in subgroups disappear or reverse when data are combined. Consequently, the cross-sectional approach was maintained to preserve the integrity of the relationships between variables and ensure the reliability of the results in the causal discovery methods.

5.4. Interpretability with Causal Graphs

The causal learning process shown in Figure 2B was implemented using three algorithms: LiNGAM, PC, and GES (Section 4.2). The application of these diverse methods enabled an exhaustive search for possible causal structures in the dataset, exploring different aspects of the relationships between variables.

To assess the consistency and significance of the causal relationships identified by the algorithms, a measure based on the entropy associated with each arc ($X_i^a \to Y_i^a$) was used. After applying the algorithms to each bootstrap sample ($S^{(1)}, S^{(2)}, \ldots, S^{(m)}$) yield causal graphs ($G^{(1)}, G^{(2)}, \ldots, G^{(m)}$). For each identified arc ($X_i^a \to Y_i^a$), its probability was calculated as ($p_{X_i^a \to Y_i^a} = \frac{1}{m}\sum_{b=1}^{m} \delta^{(b)}_{X_i^a \to Y_i^a}$), where $\delta^{(b)}_{X_i^a \to Y_i^a} = 1$ if the arc $X_i^a \to Y_i^a$ is present in the graph $G^{(b)}$, and $= 0$ otherwise.

The entropy $H(X_i^a \to Y_i^a)$ associated with each arc ($X_i^a \to Y_i^a$) was calculated as $-\left[p_{X_i^a \to Y_i^a} \log p_{X_i^a \to Y_i^a} + (1 - p_{X_i^a \to Y_i^a}) \log(1 - p_{X_i^a \to Y_i^a})\right]$, where $p_{X_i^a \to Y_i^a}$ is a Bernoulli random variable. A low value of H indicates that the presence or absence of the arc is consistent across the bootstrap samples, suggesting greater reliability and significance in the inference of that causal relationship. This approach allows to quantify the uncertainty associated with each arc, providing a solid foundation for the interpretation of the dependencies among the operational variables identified by the algorithms.

Analyzing the extracted DAG (see Figure 7) allows to identify several significant causal relationships among the operational variables. The complexity level influences the number of requirements, which affects the number of test cases which in turn determines the necessary testing hours. Furthermore, the number of testers directly impacts the testing hours which in turn influence the number of required meetings. These connections allow to understand how the monitored variables interact in the test execution process, reflecting the underlying causal structure identified through the applied algorithms.

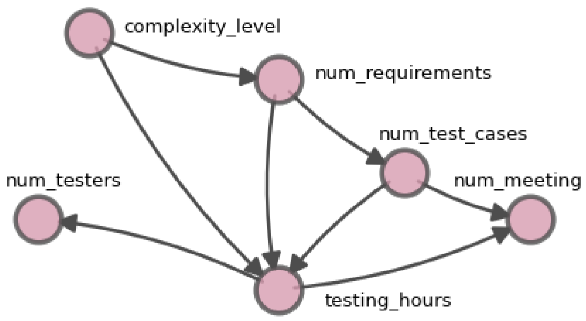

Figure 7. DAG extracted in the *Test Design* activity and monitored in the *Test Execution* activity.

According to the correlation heat map presented in Figure 8, which uses Spearman's correlation due to the nonlinearity present in some data relationships, it is observed that not all variables in Table 3 exhibit significant correlations, although some demonstrate direct causal relationships. For instance, between *num_requirements* and *num_test_cases*, a moderate correlation of 0.28 is observed, despite there being a direct causal relationship between them. Similarly, variables such as *testing_hours* and *num_testers* show a positive correlation of 0.53, supporting their causal relationship. However, in the case of *complexity_level*, a causal influence is identified on the number of requirements, but the correlation is low (0.15), illustrating that a low correlation does not imply the absence of causality.

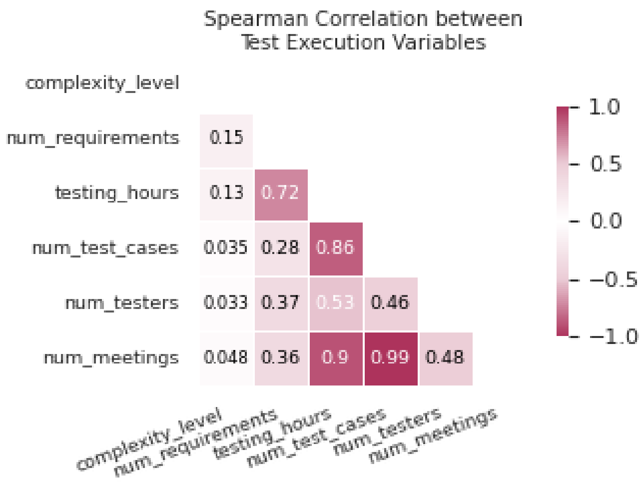

Figure 8. Correlations of the **DAG** variables in the *Test Execution* activity.

5.5. Explainability of Anomaly Attribution through Causal Graph

At this stage, as described in Section 4.3.1, the causal graph obtained during the planning phase (see Figure 7) was monitored throughout the execution. A threshold of 3σ (*three standard deviations*) was applied to detect potential anomalies in key variables, allowing for the identification of values that significantly deviated from expected outcomes. Variables exceeding this threshold were flagged as anomalous events, aiding in the identification of specific deviations during project execution. Among the cases identified using this approach are the deviations listed in Table 4. The values recorded during execution, when compared to the distribution estimated from the planning phase (see Figure 9), revealed significant deviations in several variables. Specifically, *num_requirements* (37), *testing_hours* (290.9), *num_test_cases* (132), and *num_meetings* (29.52) exceeded the threshold of three standard deviations. These anomalies, detected through the monitoring of the causal graph, indicate that project execution deviated considerably from the original plan, likely due to unforeseen complexities or changes in scope.

The violin plots presented in Figure 9 clearly show that these anomalous values were situated outside the estimated distributions.

Table 4. Anomalous values of a *Test Execution* requirement versus its *Test Design* planned value.

Variable	Anomalous Values
complexity_level	2.0
num_requirements	37.0
testing_hours	290.90
num_test_cases	132.0
num_testers	6.0
num_meetings	29.52

Figure 9. The stars (★) represent outlier values in *Test Execution* variables that exceeded (3σ).

At this analysis stage, the anomaly attribution DAG-based algorithm (see Figure 7) identified the causal topology for the observed value of *num_meetings* (Y_i^a). This enabled monitoring of the relationships among variables during the DAG execution, evaluating perturbations in the error terms (ϵ_i) for each variable, and quantifying the perturbations impact on the anomalous behavior of Y_i^a.

Decomposing the system to understand how interactions among upstream nodes impact to the final outcome is crucial to calculate the probability that Y_i^a exceeded the 3σ reference threshold.

The score attribution graph Figure 10 show how variables *num_requirements* (37) and *num_test_cases* (132) had a causal impact on the observed anomaly in *num_meetings* (29.52), exceeding the deviation thresholds. The direct impact of perturbations in these key variables showed that *num_requirements* achieved an attribution score close to 4.5, reflecting its predominant influence. *num_test_cases* had a score close to 4, also a significant impact albeit somewhat lower. In contrast, *complexity_level* (2) and *num_meetings* presented low scores of around 0 and 0.5, respectively, indicating that their influence was marginal and mediated by other variables.

The anomaly attribution analysis focused on the observed behavior of *num_meetings* (Y_i = 29.52). By applying attribution techniques, perturbations in the error terms (ϵ_{H_i}, ϵ_{C_i}, and ϵ_{Y_i}) were evaluated to examine how these variations affected the probability of Y_i exhibiting an anomalous value. In the context of the causal graph, upstream variables such as *testing_hours* (H_i = 290.90) and *complexity_level* (C_i = 2) were considered. However, their contributions to the anomaly were significantly smaller than those of *num_requirements* (R_i = 37.0) and *num_test_cases* (N_i = 132). This analysis allowed to quantify the contribution of variations in ϵ_i to the anomalous behavior of *num_meetings*, by calculating $-\log P(Y_i \geq g(Y_i) \mid \text{replacement of } \epsilon_i \text{with random values})$, where $g(Y_i)$ represents the reference threshold to determine whether Y_i is anomalous.

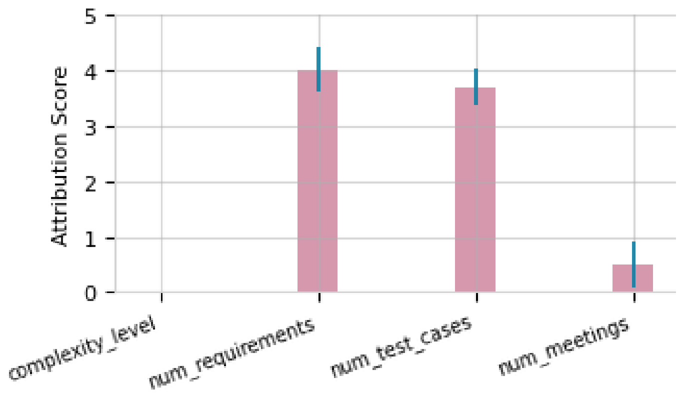

Figure 10. Attribution score of the anomalous values.

The results confirmed that *num_requirements* (R_i = 37.0) was the primary trigger, generating a cascade effect that significantly impacted other variables connected to its downstream nodes in the *Test Execution* activity during project execution.

5.6. Explainability of Distributional Changes through Causal Graph

An analysis of distribution changes in a representative subset of the data showed a significant discrepancy in *testing_hours* between the planned and observed values (see Figure 11). The planned distribution had a mean close to 150 h but the observed distribution (*execution*) had a central concentration around 250 h. This shift was confirmed with a *t*-statistic of (−11.60), evidencing a statistically significant difference in the analyzed sample. Further analysis showed that the testing times executed in this subset considerably exceeded the planned estimates, indicating a potential impact on system performance. The technique (Section 4.3.2) quantified this deviation, highlighting the need to identify the underlying causes in the DAG [11].

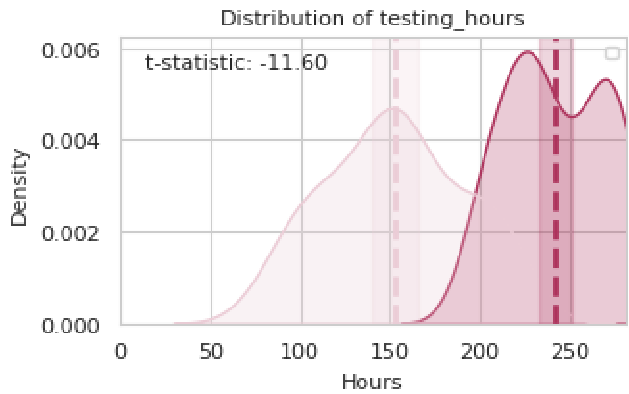

Figure 11. Distribution of *testing_hours* between the 37 Specific and Planned Cases.

The application of the distributional causal analysis technique to *testing_hours* involved comparing the conditional distributions of $P(Y_i^a \mid X_i^a)$ (*planning*) and $\hat{P}(Y_i^a \mid X_i^a)$ (*execution*). *testing_hours* (Y_i^a) was identified as the variable of interest, with parent nodes (X_i^a) including *num_test_cases*, *complexity_level*, *num_requirements*, and *num_meetings*. $P(Y_i^a \mid X_i^a)$ was estimated from the planning data and replaced by $\hat{P}(Y_i^a \mid X_i^a)$ using the data observed during execution. This produced a new distribution ($\hat{P}(T^{X_n})$), which allowed for observation of how variations in the parent nodes affected the total testing time. The KL divergence ($D_{KL}(P(Y_i^a \mid X_i^a) \parallel \hat{P}(Y_i^a \mid X_i^a))$) quantified the impact of these changes, allowing for the identification of the main causes of the discrepancy in *testing_hours*.

The analysis results show that *num_test_cases* was the primary cause of the distribution shift of *testing_hours*, with an observed difference of 250 h during execution. The Comparison of distributions confirmed a significant alteration, highlighting discrepancies between the planning and execution of the projects.

The bar chart reflects the contribution of each variable to the change in the distribution of *testing_hours* (Figure 12). *num_test_cases* has an attribution score close to 10, indicating a causal responsibility with respect to the observed change. In contrast, other variables *complexity_level*, *num_requirements*, and *num_testers* present attribution scores near zero, confirming that their effects on the distributional shift were minimal. These results highlight that the most significant change occurred in *testing_hours*, driven primarily by the causal factors represented in *num_test_cases*, while the other variables had a marginal influence.

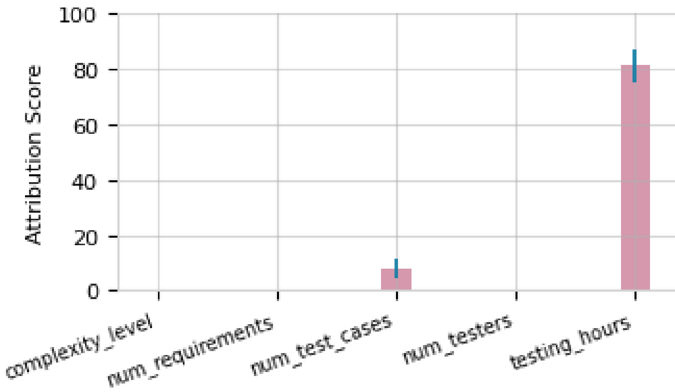

Figure 12. Attribution Score for the Distribution Change of *testing_hours*.

6. Discussion

The study of the process flow (see Figure 6) demonstrated the way in which activities recorded in the projects can be modeled as a business process flow using process mining algorithms. This representation enabled the precise identification of activities and their interactions, emphasizing the importance of the *Test Execution* activity for managers due to the outsourcing of tasks. The combined application of causal discovery algorithms, with connections evaluated through entropy (Section 5.4), facilitated the creation of a reliable causal graph, represented as a DAG (see Figure 7), which showed the causal relationships between operational variables (Section 5.3).

The anomalous values (Table 4) identified in key process variables were linked to the proposed research questions (see Section 2.3). The resulting causal graph enabled the examination of these questions through the analysis of causal pathways. The path of *num_requirements* → *testing_hours* → *num_meetings* showed the propagation of anomalies in the system (RQ.2). The graph facilitated the assessment of how specific variables influenced the observed anomalies (RQ.1), for example, the effect of *num_requirements* on *testing_hours*. This approach also allowed for the analysis of changes in the distribution of *testing_hours* (RQ.3 and RQ.4) in a subset of 37 projects, revealing how changes in causal relationships impacted the *Test Execution* activity during process execution.

Unlike previous studies [13,14,19,20], *CaProM* incorporated causal discovery into process planning. It not only identified causal relationships but also determined the influence of operational variables on anomalies and distributional changes in process activities. The integration of causality into project planning allowed addressing the dynamic nature of the environment, facilitating adjustments based on causal evidence. The causal analysis employed cross-sectional data, avoiding the need to average multiple time instances and reducing potential biases. Unlike time-series methods (e.g., [21]), this approach better preserves the relationships among variables. By avoiding issues with simultaneous data (Sections 5.3 and 5.4), the causal discovery is beneficial to process managers, minimizing spurious relationships and providing actionable information for decision making.

The results in (Sections 5.5 and 5.6) show that causal anomalies were identified in the *Test Execution* activity during the execution phase, which was monitored because of the outsourced tasks. The attribute anomalies in *num_requirements* (37) and *num_test_cases* (132) exceeded the 3σ threshold, affecting execution and contributing to the anomaly in *num_meetings* (29.52). In terms of distributional anomalies, *testing_hours* exhibited a significant shift (planned average: 150 h; observed: 250 h). Causal analysis attributed this change primarily to *num_test_cases*, with a score close to 10, while *complexity_level*, *num_requirements*, and *num_testers* made minimal contributions.

This work still has some limitations. The primary limitation (and requirement) is the fundamental need for detailed records of process planning, which are crucial for causal learning; this restricts its applicability in contexts where planning data are unavailable or incomplete. Also, the omission of organizational external factors may affect the accuracy of the causal relationships. Finally, the use of large data volumes (i.e. event logs) introduces computational challenges of its own.

7. Conclusions

This article has presented applied research aimed at improving the monitoring and interpretation of anomalous events in business process management through the early integration of causal discovery techniques. It introduced *CaProM*, an approach that uses knowledge of causal structures from initial project stages to enable monitoring and greater explainability during process execution. The key findings of this study are:

- Successful adaptation of causal discovery algorithms (LiNGAM, PC, and GES) to identify causal structures in 562 activity flow plans during the planning phase (see Section 5.1), revealing relationships not evident with traditional methods;
- Practical application of causal attribution techniques for anomalies and distributional changes, using the causal knowledge obtained during planning to more accurately interpret anomalous events during the execution of business process; and
- Demonstration of how integrating causal learning from the planning phase can significantly enhance the interpretability and explainability of anomalies during process execution, enabling a more informed and effective response.

CaProM advances the adaptation of causal learning techniques for monitoring business processes, promoting a shared understanding of project managers and process managers. It enhances analysis during execution, drawing on causal knowledge obtained during planning, and providing a solid foundation for the causal detection and explanation of anomalies and for making informed decisions. *CaProM* introduces a new way to anticipate causal scenarios in complex processes, enabling organizations to monitor them more effectively and make informed decisions during their execution. By integrating a causal view from the planning phase, managers can detect anomalies and understand their root causes using existing causal knowledge. Additionally, *CaProM* can interpret distributional changes to enable dynamic process adaptation, providing a competitive advantage in dynamic business environments that demand rapid detection, explanation, and response to changes in process patterns [26].

Future research will address unobserved confounding variables and explore the grouping of variables by process instance type, which could generate multiple **DAGs** per activity. This approach would offer a more detailed view of operational dynamics in numerous business contexts.

Author Contributions: Conceptualization, F.M. and H.A.; Methodology, F.M. and H.A.; Software, E.B.; Validation, F.M., H.A., D.D. and E.B.; Formal analysis, F.M. and H.A.; Investigation, F.M. and H.A.; Resources, H.A. and D.D.; Data curation, F.M., D.D. and E.B.; Writing—review & editing, F.M., H.A., D.D. and E.B.; Visualization, F.M., D.D. and E.B.; Supervision, H.A.; Project administration, H.A. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Data Availability Statement: The data used in this study are not publicly available due to confidentiality agreements and legal restrictions with the company that provided the information.

Conflicts of Interest: Author Fernando Montoya was employed by the company Nexus Payment Systems SpA. Author Daniela Díaz was employed by the company IT Solution SpA. The remaining authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

References

1. Pérez-Álvarez, J.M.; Maté, A.; Gómez-López, M.T.; Trujillo, J. Tactical Business-Process-Decision Support based on KPIs Monitoring and Validation. *Comput. Ind.* **2018**, *102*, 23–39. [CrossRef]
2. Márquez-Chamorro, A.E.; Resinas, M.; Ruiz-Cortés, A. Predictive monitoring of business processes: A survey. *IEEE Trans. Serv. Comput.* **2017**, *11*, 962–977. [CrossRef]
3. Jalayer, A.; Kahani, M.; Pourmasoumi, A.; Beheshti, A. HAM-Net: Predictive Business Process Monitoring with a hierarchical attention mechanism. *Knowl.-Based Syst.* **2022**, *236*, 107722. [CrossRef]
4. Cuzzocrea, A.; Folino, F.; Guarascio, M.; Pontieri, L. Predictive monitoring of temporally-aggregated performance indicators of business processes against low-level streaming events. *Inf. Syst.* **2019**, *81*, 236–266. [CrossRef]
5. Kubrak, K.; Milani, F.; Nolte, A.; Dumas, M. Prescriptive process monitoring: Quo vadis? *PeerJ Comput. Sci.* **2022**, *8*, e1097. [CrossRef] [PubMed]
6. Weinzierl, S.; Dunzer, S.; Zilker, S.; Matzner, M. Prescriptive business process monitoring for recommending next best actions. In Proceedings of the International Conference on Business Process Management, Online, 13–18 September 2020; Springer: Cham, Switzerland, 2020.
7. van de Ven, M.; Lara, Machado, P.; Athanasopoulou, A.; Aysolmaz, B.; Turetken, O. Key performance indicators for business models: A systematic review and catalog. *Inf. Syst. E-Bus. Manage.* **2023**, *21*, 753–794. [CrossRef]
8. Dawid, P. What Is a Causal Graph? *Algorithms* **2024**, *17*, 93. [CrossRef]
9. Vowels, M.J.; Camgoz, N.C.; Bowden, R. D'ya Like DAGs? A Survey on Structure Learning and Causal Discovery. *ACM Comput.* **2022**, *55*, 1–36. [CrossRef]
10. Budhathoki, K.; Minorics, L.; Blöbaum, P.; Janzing, D. Causal structure-based root cause analysis of outliers. In International Conference on Machine Learning, Baltimore, MD, USA, 17–23 July 2022.
11. Budhathoki, K.; Janzing, D.; Bloebaum, P.; Ng, H. Why did the distribution change? In Proceedings of the 24th International Conference on Artificial Intelligence and Statistics, Virtual, 13–15 April 2021.
12. Tian, J.; Pearl, J. Causal discovery from changes. In *Proceedings of the 17th Conference in Uncertainty in Artificial Intelligence*. Morgan Kaufmann Publishers Inc.: San Francisco, CA, USA, 2001; pp. 512–521.
13. Shoush, M., Dumas, M. Prescriptive process monitoring under resource constraints: A causal inference approach. In Proceedings of the International Conference on Process Mining, Eindhoven, The Netherlands, 31 October–4 November 2021; Springer International Publishing: Cham, Switzerland, 2021; pp. 180–193.
14. Bozorgi, Z.D.; Dumas, M.; Rosa, M.L.; Polyvyanyy, A.; Shoush, M.; Teinemaa, I. Learning When to Treat Business Processes: Prescriptive Process Monitoring with Causal Inference and Reinforcement Learning. In *Advanced Information Systems Engineering*; Indulska, M., Reinhartz-Berger, I., Cetina, C., Pastor, O., Eds.; In Proceedings of the CAiSE 2023, Zaragoza, Spain, 12–16 June 2023; Springer: Cham, Switzerland, 2023.
15. Mehdiyev, N.; Majlatow, MFettke, P. Quantifying and explaining machine learning uncertainty in predictive process monitoring: An operations research perspective. *Ann. Oper. Res.* **2024**, 1–40.. [CrossRef]
16. Wang, J.; Chang, V.; Yu, D.; Liu, C.; Ma, X.; Yu, D. Conformance-oriented Predictive Process Monitoring in BPaaS Based on Combination of Neural Networks. *J. Grid Comput.* **2022**, *20*, 25. [CrossRef]
17. Kotsias, S.; Kerasiotis, A.; Bousdekis, A.; Theodoropoulou, G.; Miaoulis, G. Predictive and Prescriptive Business Process Monitoring with Reinforcement Learning. In Proceedings of the Novel Intelligent Digital Systems, 2nd International Conference (NiDS 2022), Athens, Greece, 29–30 September 2022; Krouska, A., Troussas, C., Caro, J., Eds.; Springer: Cham, Switzerland, 2022.
18. Bozorgi, Z.D.; Teinema, I.; Dumasa, M.; Rosa, M.L.; Polyvyanyy, A. Prescriptive process monitoring based on causal effect estimation. *Inf. Syst.* **2003**, *116*, 102198. [CrossRef]
19. Delias, P.; Mittas, N.; Florou, G. A doubly robust approach for impact evaluation of interventions for business process improvement based on event logs. *Decis. Anal. J.* **2023**, *8*, 100291. [CrossRef]
20. Brunk, J.; Stierle, M.; Papke, L.; Revoredo, K.; Matzner, M.; Becker, J. Cause vs. effect in context-sensitive prediction of business process instances. *Inf. Syst.* **2021**, *95*, 101635. [CrossRef]
21. Hompes, B.F.A.; Maaradji, A.; La Rosa, M.; Dumas, M.; Buijs, J.C.A.M.; van der Aalst, W.M.P. Discovering Causal Factors Explaining Business Process Performance Variation. In *Lecture Notes in Computer Science, Proceedings of the Advanced Information Systems Engineering, CAiSE 2017, Essen, Germany, 12–16 June 2017;* Dubois, E., Pohl, K., Eds.; Springer: Cham, Switzerland, 2017.
22. Zanga, A.; Ozkirimli, E.; Stella, F. A Survey on Causal Discovery: Theory and Practice. *Int. J. Approx. Reason.* **2022**, *151*, 101–129. [CrossRef]
23. Zhou, W.; Chen, Q. A Survey on Causal Discovery. In *Knowledge Graph and Semantic Computing: Knowledge Graph Empowers the Digital Economy*; Sun, M., Qi, G., Liu, K., Ren, J., Xu, B., Feng, Y., Liu, Y., Chen, Y., Eds.; Springer Nature: Singapore, 2022.

24. Wang, L.; Huang, S.; Wang, S.; Liao, J.; Li, T.; Liu, L. A survey of causal discovery based on functional causal model. *Eng. Appl. Artif. Intell.* **2024**, *133*, 108258. [CrossRef]
25. Augusto, A.; Conforti, R.; Dumas, M.; La Rosa, M. Split miner: Discovering accurate and simple business process models from event logs. In Proceedings of the IEEE International Conference on Data Mining, New Orleans, LA, USA, 18-21 November 2017.
26. Zyphur, M.J.; Bonner, C.V.; Tay, L. Structural equation modeling in organizational research: The state of our science and some proposals for its future. *Annu. Rev. Organ. Psychol. Organ. Behav.* **2023**, *10*, 495–517. [CrossRef]

4.5 Causal Process Mining for Temporal Deviations in Business Event Flows

This article brings together the elements developed previously and applies them to a complete industrial workflow for activity planning. The study is situated at the interventional level of the hierarchy (**L**$_2$) and combines disaggregation by variants, clustering of temporal deviations, and causal discovery to quantify the impact of critical activities on process performance (**SO.1**, **SO.3**; **RQ.1**, **RQ.3**). With this step, the contribution ceases to be limited to isolated cases and is presented as a general causal process mining procedure for temporal deviations, which the literature can use as a basis for evaluating interventions on performance.

Publication Summary

Title:	*Causal Process Mining for Temporal Deviations in Business Event Flows*
Authors:	Fernando Montoya, Gonzalo Rivera-Lazo, Mauricio Hidalgo and Hernán Astudillo
Conference:	Applied Industrial Engineering: Intelligent Models and Data Engineering (CIIA)
Publisher:	E3S
Year:	2025
Doi:	https://doi.org/10.1051/e3sconf/202565801006

Causal Process Mining for Temporal Deviations in Business Event Flows

Fernando Montoya[1,2],[*], *Gonzalo* Rivera-Lazo[3],[**], *Mauricio* Hidalgo[4],[***], and *Hernán* Astudillo[5],[****]

[1]Universidad Técnica Federico Santa María, Santiago, Chile
[2]Fundación Instituto Profesional Duoc UC, Santiago, Chile
[3]Corriente Sur, Valparaíso, Chile
[4]Facultad de Ingeniería, Universidad San Sebastián, Santiago, Chile.
[5]ITiSB - Universidad Andrés Bello, Viña del Mar, Chile

Abstract. Efficient management of business processes is crucial in dynamic environments. Temporal deviations in execution times can affect workflows. They also increase costs and reduce service quality. Traditional process analysis methods generally rely on correlations. They do not establish causal relationships that explain these deviations. Existing approaches face challenges in identifying the underlying causes of temporal deviations. This difficulty limits the ability to apply precise interventions. Without a causal understanding, corrective actions may prove ineffective. They often address symptoms rather than underlying causes. This study integrates process mining with causal inference. The objective is to explain the factors that generate temporal deviations in enterprise workflows at the variant level. Causal discovery techniques are applied (`LiNGAM`, `PC`, and `GES`). Event logs are grouped by variant and stratified into temporal-deviation clusters. This procedure identifies critical activities. Directed acyclic graphs (`DAG`) are also recovered to quantify their impact on delays. In addition, alternative scenarios are simulated using the $\mathtt{do}(\cdot)$ operator. These interventions provide actionable information for optimization. The methodology was validated with a dataset of 562 planning instances from a payment-processing company. The results identified causal relationships consistent with the learned `DAG`s. Among them, the impact of scheduled hours and complexity level on execution times. This work integrates process mining and causal inference. The result is a framework for data-driven decision-making in operational process management. The proposal improves the interpretability of temporal deviations. It also enables the design of more precise interventions. It contributes to establishing a methodological foundation for continuous improvement through causal analysis.

1 Introduction

In dynamic, competitive environments [1], organizations must manage their business processes effectively to achieve strategic objectives. Controlling operational performance is es-

[*]e-mail: fernando.montoya@usm.cl
[**]e-mail: gonzalo@corrientesur.com
[***]e-mail: mauricio.hidalgob@uss.cl
[****]e-mail: hernan@acm.org

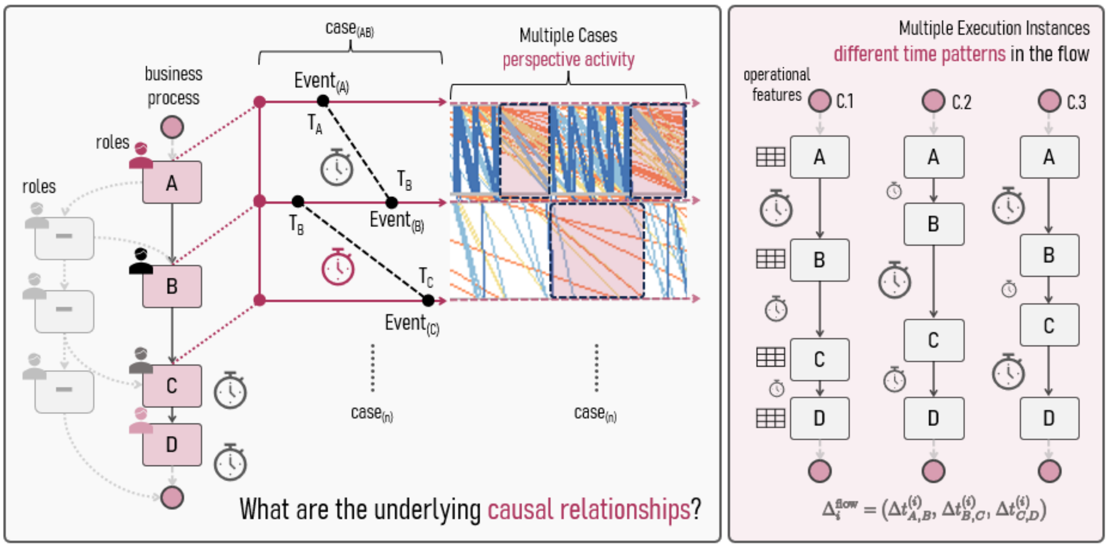

Figure 1. Visual characterization of temporal deviations and their possible underlying causal relationships in business processes.

sential, particularly the execution times of activities. Within this framework, process mining, complemented by causal inference techniques, provides tools to identify and understand deviations.

As shown in Figure 1, temporal deviations appear in repeated executions of the same process. Multiple nominally identical instances are observed, but with distinct temporal patterns. This variability makes it difficult to identify and explain causal relationships, especially when roles, activities, and operational characteristics of the process interact.

Despite detailed planning of activities and schedules, temporal deviations persist [2]. Their effects are adverse: they increase operating costs, introduce delays, and reduce service quality. They also affect collaboration among roles [3]. Traditional methods rely on correlations [4, 5]. They do not clearly establish cause-and-effect relationships. This limitation reduces the ability to implement effective and timely corrective actions [6].

To address this problem, the present work proposes an integrated methodology. It combines process mining [10] with causal machine learning techniques [7]. The main objective is to identify and analyze cause-and-effect relationships associated with temporal deviations [8, 9]. The analysis operates at the variant level and uses stratification into temporal-deviation clusters. This favors interpretability and enables the design of interventions.

The specific objectives are: (***i***) to integrate process mining with advanced causal machine learning techniques; (***ii***) to identify causal relationships that explain temporal deviations in business processes; and (***iii***) to provide a framework for well-founded and timely managerial interventions, supported by clear and robust causal analysis.

In line with the above, this article poses the following research questions:

1. $\boldsymbol{RQ_1}$ How can process mining be integrated with causal machine learning techniques to identify specific causes of temporal deviations at the variant and cluster levels?

2. $\boldsymbol{RQ_2}$ What are the key causal relationships associated with the temporal deviations observed in business processes?

3. $\boldsymbol{RQ_3}$ To what extent does causal understanding improve the quality and effectiveness of managerial decisions and interventions in the face of temporal deviations?

The organization of the document is as follows. *Preliminaries* presents the key concepts (2). *Related Work* reviews prior studies (3). *Proposed Methodology* describes the approach based on process mining and causal inference (4). The *Results* are then reported (5), followed by the *Discussion* (6) and the *Conclusions* (7).

2 Preliminaries

Below, the essential concepts and notations that underpin the causal analysis developed in this study are presented, along with their respective formal definitions.

Definition 2.1 (Business Process Model)**.** Let a business process model be defined as the tuple $P = (\mathcal{A}, \prec)$, where $\mathcal{A}$ is a finite set of activities and $\prec$ establishes the precedence relationship among them. Each case $c \in C$ generates a trace σ, represented by the sequence of executed activities with their respective timestamps $\sigma_i = ((a_1, t^i_{a_1}), (a_2, t^i_{a_2}), \ldots, (a_n, t^i_{a_n}))$.

Here, $t^i_{a_k}$ is the timestamp of a_k in the case c_i. The set of all traces is denoted by Σ. A process variant is an equivalence class over Σ, determined by the sequence of activities. The projection operator act is defined as $\text{act}(\sigma_i) = \langle a_1, a_2, \ldots, a_n \rangle$. Two traces σ_p, σ_q belong to the same variant v_j if and only if $\text{act}(\sigma_p) = \text{act}(\sigma_q)$. The set of variants is denoted by $\mathcal{V} = \{v_1, v_2, \ldots, v_k\}$.

Definition 2.2 (Temporal Deviation)**.** The time interval between two consecutive activities a_j, a_{j+1} in σ_i is given by $\Delta t^i_{a_j a_{j+1}} = t^i_{a_{j+1}} - t^i_{a_j}$. The temporal deviation is obtained by comparing each $\Delta t^i_{a_j a_{j+1}}$ with the average interval $\overline{\Delta t_{a_j a_{j+1}}}$, forming the set $\Delta_{a_j a_{j+1}} = \{\Delta t^i_{a_j a_{j+1}} - \overline{\Delta t_{a_j a_{j+1}}} : i = 1, \ldots, m\}$.

Definition 2.3 (Average Causal Effect (ATE))**.** It is the expected average change in Y when, for the same population, X is exogenously set to (x') instead of (x); it is written as $\text{ATE}(x', x) = \mathbb{E}[Y \mid do(X = x')] - \mathbb{E}[Y \mid do(X = x)]$ or, in potential-outcomes terms, $\mathbb{E}[Y(x') - Y(x)]$ with $Y_{(x)}$ denoting the outcome that a unit would have if X were set to x. The observational difference $(\mathbb{E}[Y \mid X = x'] - \mathbb{E}[Y \mid X = x])$ compares groups that naturally take those values of X and may reflect other differences between them; the ATE answers what would occur under an exogenous manipulation of X.

Definition 2.4 (Causal Intervention)**.** A $do(X = x)$ intervention externally sets X to the value x, severing all influences pointing into X; in structural models it is equivalent to replacing the structural equation for X with the assignment $X := x$, and in a DAG to removing the incoming edges to X. The distribution $P(Y \mid do(X = x))$ describes the behavior of Y under that manipulation and typically differs from $P(Y \mid X = x)$ because the latter reflects who has $X = x$ in the observed data, not the result of forcing $X = x$.

3 Related Work

3.1 Temporal Deviations in Business Processes

Recent work on temporal deviations in business processes has followed two main lines: on the one hand, the detection and representation of timing patterns; on the other, the search for causal relationships to guide intervention decisions.

Within the family of temporal signatures, Böhmer and Rinderle-Ma [14] compare historical and current process behavior to enable early anomaly detection. Janina et al. [15] extend this idea with signatures centered on the intervals between consecutive activities. These studies increase sensitivity to temporal changes, although their scope narrows as trace length and the structural complexity of the process grow.

Regarding the representation of constraints, Pereira and Varajão [12] introduce specific temporal links in modeling languages. Senderovich et al. [13] propose the Temporal Network Representation (`TNR`) to capture relationships between activities with probabilistic support. Zhao et al. [18] present a temporal hierarchical model (`TH-BPM`) that organizes these constraints at different levels. These proposals offer an expressive framework; nevertheless, variability across instances and scalability remain challenges in scenarios with pronounced heterogeneity.

In operational monitoring and uncertainty management, Rogge-Solti and Kasneci [11] apply Bayesian methods to handle noise and complex dependencies in dynamic contexts. Richter and Seidl [16] introduce `TESSERACT` for real-time monitoring via exponential moving averages, and Asma et al. [20] incorporate metrics such as latency and temporal capacity. These techniques facilitate continuous process surveillance, although their performance depends on data quality and may degrade under abrupt changes.

Another line relies on data structures for anomaly detection. Mavroudopoulos and Gounaris [17] use M-trees and R-trees to locate deviations at different granularities. Richter et al. [19] explore proximity-based variants that report accuracy improvements over simpler approaches, with difficulties that increase in high-dimensional settings.

3.2 Causality in Business Processes

Qafari and Van der Aalst [21, 24] propose structural equation models integrated with algorithms such as `GFCI` to identify root causes in performance and compliance issues. In a similar vein, Luo et al. [22] combine process mining and causal discovery through the `ACLP` framework and the `SMMB` algorithm, focusing on identifying causes in complex and dynamic processes. Both approaches stand out for offering effective methods to identify root causes, although they face common limitations related to reliance on expert knowledge and the difficulty of handling high variability in complex structures.

4 Proposed Method

The proposed methodology, illustrated in Figure 2, addresses the causal analysis of temporal deviations by focusing on specific variants of the business process. This approach differs from previous methods mentioned in Section 3.1, which analyze all log traces together. By studying variants individually, Simpson's paradox, where clear trends in subgroups disappear or change when all data is combined, is avoided. In this way, precise causal relationships are identified, aligning with the specific strategic objectives of each process variant. This method reduces biases and interferences typical of general approaches such as those by Böhmer and Rinderle-Ma [14] and Janina et al. [15], who use general signatures and may encounter distortions when considering multiple variants simultaneously.

The following stages are closely related to addressing this problem. First, the individual process variants are identified, and then robust causal analysis is applied to each variant, ensuring precise and contextualized causal identification.

4.1 Event Log From Planning:

The first stage of the methodology, illustrated in Figure 2, focuses on identifying the business process model using process mining [25]. For this purpose, event logs (L) derived from previously defined project plans are obtained. An event log is formally represented as a set of traces σ, where each trace σ_i consists of ordered sequences of activity-time pairs $\sigma_i = \{(a_1, t^i_{a_1}), \ldots, (a_n, t^i_{a_n})\}$, following the notation presented in the preliminary section.

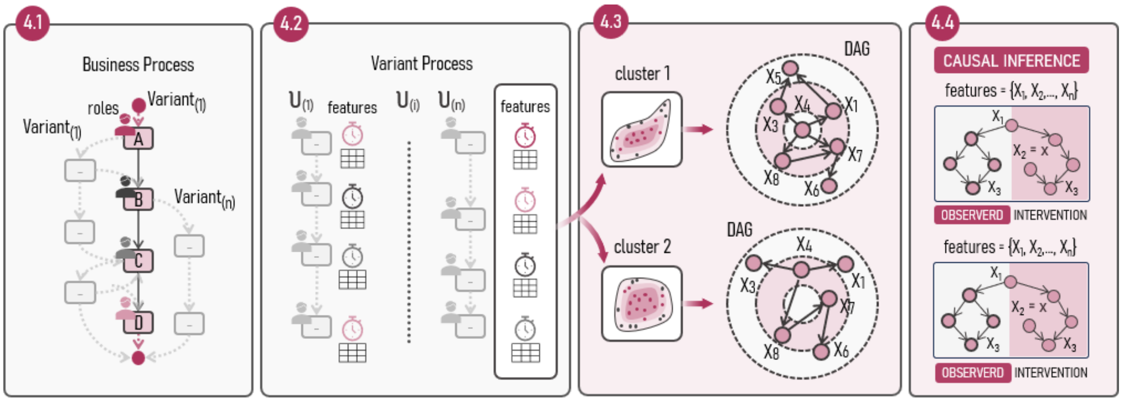

Figure 2. Scheme of the proposed methodology for analyzing underlying causal relationships in temporally clustered deviations from a process variant.

This strategy allows capturing the expected behavior of the process according to the original planning. Moreover, it enables the early detection of temporal deviations and the recognition of interactions among the roles involved in the process, as suggested by Pereira and Varajão [12]. This early detection is essential for understanding the causes of the deviations before they negatively affect the final outcomes. In this way, it contributes to a better understanding and management of the temporal and collaborative relationships among roles within the process flow.

Table 1. Event Log Extract

Case Id	**Activity**	**Time-span**	**Role**$_1$	**...**	**Feature**$_n$
case 1	A	10.3 ms	role.$_1$	...	feat.$_2$
case 1	B	45.2 ms	role.$_2$	...	feat.$_3$
case 1	C	62.5 ms	role.$_4$	...	feat.$_6$
case 2	A	72.4 ms	role.$_5$	...	feat.$_3$
case 2	E	43.8 ms	role.$_1$	...	feat.$_2$
...	...	...	...	...	...

In Table 1, a typical extract of the event log obtained from the project plans is shown. Each record presents the following essential attributes:

1. **Case ID**: Unique identifier that distinguishes each process instance. *Examples* (`ERP/CRM`): `C-2025-00451`, `OP-789231`, `TICKET-31415`.
2. **Activity**: Activity performed at a point in the process. *Examples:* Request receipt, Data validation, Credit approval, Generate invoice, Close case.
3. **Timestamp(s)**: Event timestamp(s) (start and/or end) to compute duration and waiting times between activities. *Example:* start 2025-06-15 09:34, end 10:12 → duration 38 min.
4. **Role**: Role responsible for executing the activity (function or profile). *Examples:* Analyst, Supervisor, Back-office; (optional) individual resource `u1234`.
5. **Additional operational variables**: Attributes that contextualize the case or the event. *Examples:* priority (*High/Medium/Low*), channel (*Web/In-person*), product/service, customer segment, amount, number of items, location, `SLA`.

This structure facilitates clearly capturing the expected behavior of the process according to the initial planning. Moreover, it allows for the early identification of temporal deviations and reveals interaction patterns among roles. These elements provide a solid foundation for subsequent causal analysis, highlighting improvements over methods such as those described in [14] and [15], which tend to rely on more general representations.

4.2 Business Process Variant:

The second stage of the methodology, in line with the definitions and notations established in the definition 2.1 (*see* Sec. 2), focuses on extracting and characterizing operational variables from each process variant. Specifically, the execution times between consecutive activities within each process variant V_j, previously defined in the Preliminary section, are extracted. These times are transformed using the `Modified Z-Score`, formally defined as:

$$Z_m(\Delta t^i_{a_j a_{j+1}}) = \frac{0.6745(\Delta t^i_{a_j a_{j+1}} - \tilde{\Delta} t_{a_j a_{j+1}})}{MAD(\Delta t_{a_j a_{j+1}})}$$

where $\Delta t^i_{a_j a_{j+1}}$ is the time interval between consecutive activities for trace σ_i, $\tilde{\Delta} t_{a_j a_{j+1}}$ represents the median time interval in variant V_j, and `MAD` corresponds to the median absolute deviation. In this way, a temporal vector $Z_j = [Z_m(\Delta t_{a_1 a_2}), Z_m(\Delta t_{a_2 a_3}), \ldots, Z_m(\Delta t_{a_{n-1} a_n})]$. is obtained, which will later be used as input for the clustering and causal graph discovery stages. Furthermore, this temporal vector forms the fundamental basis for clearly identifying and analyzing the specific temporal deviations of each process variant, thereby enabling a precise understanding of the underlying causal relationships. vspace-0.1cm

4.3 Cluster of Temporal Deviations:

The third stage of the proposed methodology (*see* Figure 2) consists of applying the Gaussian Mixture Model (`GMM`) to the temporal vectors normalized using the `Modified Z-Score`, previously described as Z_j. This vector serves as the input to the GMM, which is an unsupervised machine learning model, formally expressed as

$$p(Z_j) = \sum_{k=1}^{K} \pi_k \mathcal{N}(Z_j \mid \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k),$$

where π_k is the weight of the k-th Gaussian component, $\boldsymbol{\mu}_k$ is the mean vector, and $\boldsymbol{\Sigma}_k$ is the covariance matrix associated with the component. The application of the model allows for the identification of clusters that group traces with similar patterns of temporal deviations.

These obtained clusters facilitate subsequent causal analysis on the associated operational variables. The identification of typical causal structures is illustrated in Figure 3, showing three basic configurations: Forks, Chains, and Colliders. Recognizing these causal patterns is crucial for interpreting the underlying dynamics in temporal deviations, which directly contributes to greater causal interpretability and the direction of the causal effect of interventions.

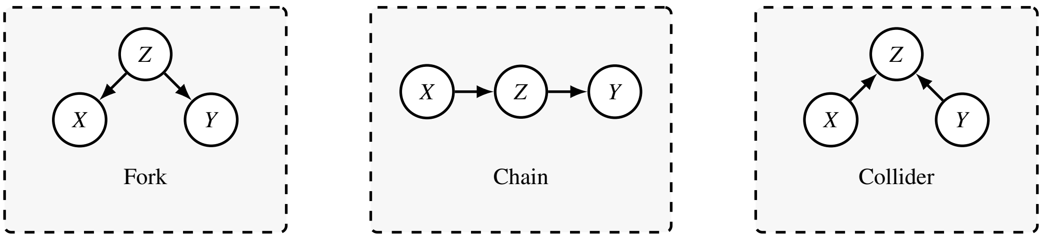

Figure 3. Causal topologies (Fork, Chain, Collider) illustrating relationships among variables.

Algorithm 1 Causal Discovery to Obtain the DAG (Condensed)

Input: $D \in \mathbb{R}^{n \times p}$ (with p variables $\{X_1, \ldots, X_p\}$);
$B \in \mathbb{N}$ (number of bootstraps);
$\tau \in [0, 1]$ (inclusion threshold).
Output: $\mathcal{G}^* = (V, E^*)$ with $V = \{X_1, \ldots, X_p\}$.

Definitions: For each $b = 1, \ldots, B$ and $X_i, X_j \in V$:

$$\delta_{ij}^{(b)} = \begin{cases} 1, & \text{if } \exists\, m \in \{\text{LiNGAM, PC, GES}\} : X_i \to X_j \text{ in } D^{(b)}, \\ 0, & \text{otherwise.} \end{cases}$$

Let $v_{ij} = \sum_{b=1}^{B} \delta_{ij}^{(b)}$, $f_{ij} = \frac{v_{ij}}{B}$, and $E_{\text{pre}} = \{(i, j) \mid f_{ij} \geq \tau\}$.
`d-sep`(X_i, X_j, Z) returns `true` if X_i and X_j are d-separated by Z in D, else `false`.

for $b \leftarrow 1$ ***to*** B **do**
  $D^{(b)} \leftarrow$ bootstrap sample of D; **foreach** $(i, j) \in V \times V$ **do**
    if $\exists\, m \in$ {LiNGAM, PC, GES} *with* $X_i \to X_j$ *in* $D^{(b)}$ **then**
      $\delta_{ij}^{(b)} \leftarrow 1$ /* *Causal edge detected in current bootstrap sample* */
    else
      $\delta_{ij}^{(b)} \leftarrow 0$ /* *No causal edge detected in current bootstrap sample* */
    $v_{ij} \leftarrow v_{ij} + \delta_{ij}^{(b)}$
foreach $(i, j) \in V \times V$ **do**
  $f_{ij} \leftarrow \frac{v_{ij}}{B}$ /* *Calculate frequency of edge occurrence* */
$E_{\text{pre}} \leftarrow \{(i, j) \mid f_{ij} \geq \tau\}$; $\mathcal{G}_{\text{pre}} \leftarrow (V, E_{\text{pre}})$ **while** $\mathcal{G}_{\text{pre}}$ *contains a cycle* C **do**
  $E' \leftarrow \{(i, j) \in C \mid$ `d-sep`$(X_i, X_j, Z) =$ `true`$\}$; **if** $E' \neq \emptyset$ **then**
    $(i^*, j^*) \leftarrow \arg\min_{(i,j) \in E'} v_{ij}$ /* *Lowest support edge selected* */; Remove (i^*, j^*) from E_{pre} /* *Edge removed to break cycle* */; Update $\mathcal{G}_{\text{pre}} \leftarrow (V, E_{\text{pre}})$ /* *Graph updated* */
  else
    break /* *No removable edge found: process terminated* */
return $\mathcal{G}^* \leftarrow \mathcal{G}_{\text{pre}}$

To obtain the causal graphs (DAG), an iterative procedure based on techniques such as LiNGAM, PC, or GES is implemented, as described in Algorithm 1. This algorithm combines statistical tests for conditional independence with cycle elimination, thereby ensuring the generation of coherent causal graphs.

The resulting DAG offers a robust interpretation of how the operational variables causally impact the detected temporal deviations, facilitating the implementation of effective strategic interventions in process management.

4.4 Interventionist Causal Design:

Once the clusters of temporal deviations have been defined and the corresponding DAG has been obtained, the interventionist causal analysis is undertaken to determine the direction of influence among the variables and to implement targeted interventions. In this model, the treatment T is understood as the action applied to correct or improve the process, the outcome

Y reflects the performance measured after the intervention, and the confounding variable Z groups the contextual factors that affect both.

The model is expressed using causal functions:

$$T := f_T(N_T), \quad Y := f_Y(T, Z, N_Y),$$

where N_T and N_Y are independent noise terms. The intervention $\mathrm{do}(T = t)$ is applied to fix T at a specific value, which produces the distribution $P_{Y|\mathrm{do}(T=t)}$. By comparing this distribution with the observational one, the direct effect of T on Y is isolated while controlling for the influence of Z.

This interventionist approach confirms the causal direction and, unlike the methods of Qafari and Van der Aalst [21, 24] and of Luo et al. [22], which relied on observational analyses without incorporating the perspective of intervention, enables the design of precise interventions. In this manner, improvement strategies can be established to optimize the flow of events within the business process.

5 Results

This section presents the findings obtained by applying the methodology described in Section.4 to an event log documenting the planning and development of technological requirements at a payment services company. A total of 562 instances were evaluated, allowing for the identification of patterns in temporal deviations and in the operational structure of the process.

5.1 Dataset Description

The analysis was conducted on an event log that records the sequence of activities, start and end times, and other operational attributes for each planning instance. Each instance documents a complete process, capturing both the planning and execution phases. Table 2 summarizes the main characteristics of the dataset.

Table 2. Dataset Characteristics

Characteristic	**Value**
Planning Instances	562
Number of Unique Activities	8
Average Events per Planning	10
Registered Operational Attributes	12
Time Period	2018–2020
Source	Event Log

5.2 Principal Findings

The process model quantifies the duration of key activities and detects critical paths (*see* Figure 4). Notable differences were observed in tasks such as *User Evaluation Clarification* (*averaging* 34.9 *days*) and *Production Testing Evaluation* (47.4 *days*), both of which have a significant impact on the workflow. Additionally, *Quotation Approval* reaches 33.4 days, while *Functional Definition* (23.7 *days*) and *Planning* (13.7 *days*) also stand out due to their complexity. On the other hand, "Technical Evaluation" records 11 days, *Quotation Generation* 7.2 days, and *Completeness Review* 5 days, reflecting relatively shorter durations. These figures highlight the heterogeneity of processing times and help identify critical paths within the process.

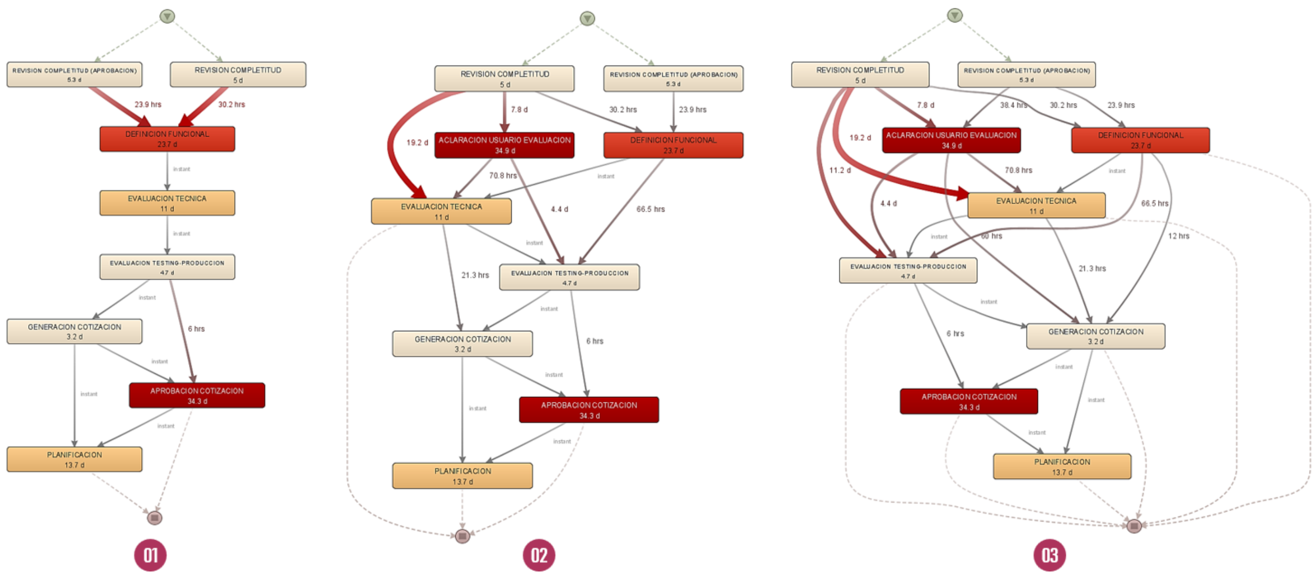

Figure 4. Three representations of the same process with different relevance filters: **(1)** activities related to `Quotation Approval` are highlighted, **(2)** flow variants that modify the sequence of activities are shown, and **(3)**a bottom-up view focused on execution times and the confluence of paths is integrated.

5.3 Temporal Deviations Clustering

To study the temporal deviations within the variant, `GMM` and `PCA` were applied to the execution times. Figure 5 shows two main classes (*class = 0 and class = 1*), with 348 and 168 instances respectively, along with 46 cases of overlap. This overlap exhibits an entropy of $H(X) = -\sum_i p(x_i) \log p(x_i) = 4.52$, indicating a moderate ambiguity in class separation. This clustering facilitates the identification of similar deviation patterns; for instance, `class=1` includes traces with greater delays in *User Evaluation Clarification* and *Quotation Approval*, suggesting a significant impact on overall planning.

5.4 Interventionist Causal Analysis

In the subgroup of *class = 1* (168 *planning instances with similar patterns*), a causal analysis focused on the activity *Clarification User Evaluation* was applied. For this purpose, the following operational variables were identified (*see* Table 3):

Table 3. Variables in "Clarification User Evaluation".

Variable	**Description**	**Role**
`scheduled_hours`	Scheduled hours.	Treatment (T)
`num_requirements`	Requirements to clarify.	Outcome (Y)
`complexity_level`	Complexity level (1 to 3).	Confounder (Z)

The causal model is defined as:

$$T := f_T(N_T),$$
$$Y := f_Y(T, Z, N_Y),$$

where T represents the scheduled hours and Y the number of requirements. N_T and N_Y are independent noise terms, while Z groups confounding factors such as the project's complexity and the number of services involved.

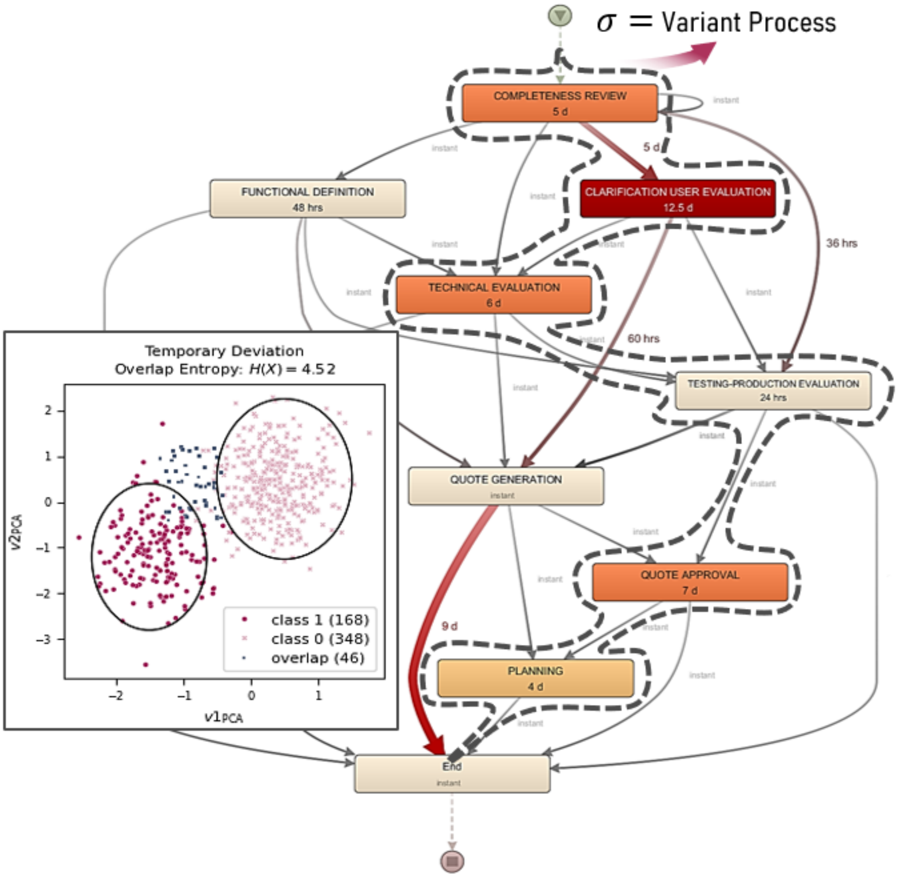

Figure 5. Process variant with segmented activities and temporal deviations clustering. Two classes (`class=0` and `class=1`) are observed, with 46 overlapping cases and an entropy of $H(X) = 4.52$.

To illustrate the analysis, the intervention do($T = 2.5$) was applied, fixing T at 2.5 hours. This allowed us to obtain the distribution $P_{Y|\text{do}(T=2.5)}$ and compare it with the observational distribution, thereby isolating the effect of T on Y. The average causal effect (ATE) was calculated as:

$$\text{ATE}(t, t') = \mathbb{E}[Y \mid do(T = t)] - \mathbb{E}[Y \mid do(T = t')],$$

yielding a value of −0.196 per additional hour. For an increase of 12 hours (from 2.5 to 14.5), the ATE was −2.355, suggesting a non-linear behavior and possible synergy effects.

The causal structure (see Figure 6) was derived using LiNGAM [29], as well as the PC and GES algorithms [30] (*see* algorithm 1). As shown in Figure 6 , *scheduled_hours* has a direct effect on *num_requirements*, while *complexity_level* acts as a confounding variable.

6 Discussion

Discussion

The analysis showed that the activities *Clarification User Evaluation* (12.5 *days*) and *Testing Production Evaluation* (24 *days*) exhibited prolonged durations. Sequences were also detected in which minor delays accumulate and create bottlenecks. These results make it possible to locate critical points in the process and guide targeted interventions to improve performance (*see* RQ.2).

The findings are consistent with research that prioritizes attention to stages with greater delays. Böhmer and Rinderle [14] use temporal signatures to detect anomalies early, whereas

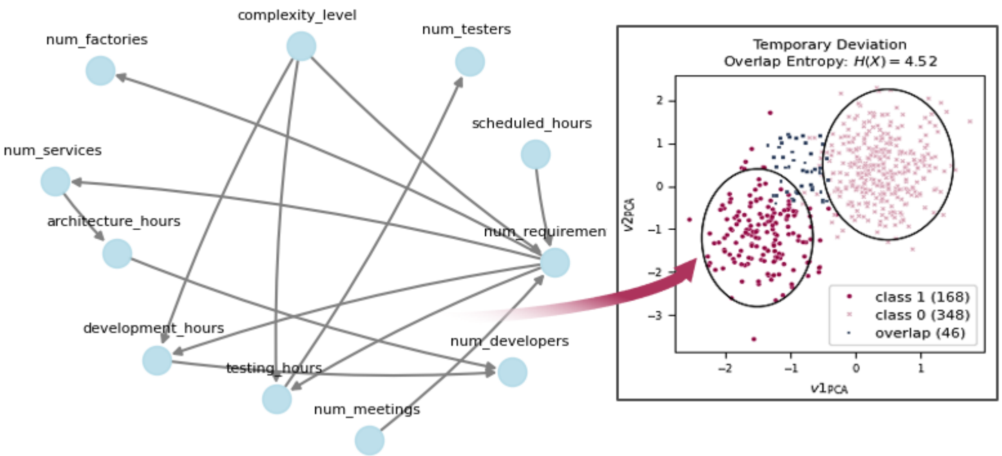

Figure 6. Causal graph depicting operational variables within the `Clarification User Evaluation` activity, with causal relationships identified using causal discovery techniques.

Janina et al. [15] incorporate detailed signatures to identify deviations between consecutive activities. Although these approaches present limitations in lengthy processes, they reinforce the need to manage critical stages effectively (*see* RQ.2).

The use of a real dataset (562 *planning instances*) and the focus on specific activities strengthen the validity of the observations. Nevertheless, the limited time window and reliance on a single dataset constrain generalization. Future studies should consider different organizational contexts and extend the observation period to assess the stability and applicability of the results (*see* RQ.1).

It is also pertinent to examine the influence of organizational factors, such as resource allocation and the adoption of automation technologies on reducing temporal variations. This would allow a deeper understanding of how specific interventions improve process efficiency (*see* RQ.3).

In summary, the precise identification of critical activities and the analysis of problematic sequences provide a clear basis for targeted interventions. Their implementation can reduce execution times and improve operational efficiency, contributing to the optimization of overall process management (*see* RQ.3).

7 Conclusions

This work integrated process mining and causal machine learning to identify and explain temporal deviations in enterprise workflows. The methodology combines causal discovery techniques with the analysis of event logs to improve the interpretability of variations in execution times.

The results indicate that certain activities are associated with delays and that their impact can be assessed through causal interventions. The estimation of the average causal effect quantified the effect of variables such as scheduled hours and complexity level on process duration. Although the approach is applicable in different settings, generalizing the findings requires testing it across diverse organizational configurations and expanding the dataset. Future work can explore the role of resource availability and automation in reducing deviations.

In summary, the incorporation of causal models into process mining provides a useful framework for optimizing operational management. Understanding cause-and-effect relationships facilitates the design of more precise interventions, leading to reduced execution times and greater efficiency in process planning.

References

[1] Mahdia, A., Boukhedouma, S., Alimazghi, Z.: Impact of Strategic Alignment on Company Performance: An approach based on performance indicators system design. In: 2020 International Conference on Advanced Computer Science and Information Systems (ICACSIS), pp. 233–240. IEEE (2020).

[2] Sontheim, J., Richter, F., Seidl, T.: Temporal Deviations on Event Sequences. (2018)

[3] Minseok, Song., Wil, M., P., van, der, Aalst. (2008). Towards comprehensive support for organizational mining.

[4] Smith, G., Cordes, J.: Confusing Correlation with Causation. In: The 9 Pitfalls of Data Science, pp. 155–172. Oxford University Press (2019).

[5] Brown, A.: Correlation and Causation. Wilmott, 2022(118), 12–15 (2022).

[6] Juegos, P.A.: Correlación y causalidad: un error lógico. Revista de Educación Experimental, 58(4), 271–279 (1990).

[7] Cui, P., Shen, Z., Li, S., Yao, L., Li, Y., Chu, Z., & Gao, J. (2020, agosto). La inferencia causal se encuentra con el aprendizaje automático. En Actas de la 26ª conferencia internacional ACM SIGKDD sobre descubrimiento de conocimiento y minería de datos (pp. 3527-3528).

[8] Zhou, W., Chen, Q.: A Survey on Causal Discovery. (2021)

[9] Yao, L., Chu, Z., Li, S., Li, Y., Gao, J., Zhang, A.: A Survey on Causal Inference. ACM Transactions on Knowledge Discovery From Data, 15(5), 1–37 (2021).

[10] Ahmed, R., Faizan, M., Burney, A.I.: Process Mining in Data Science: A Literature Review. In: 2019 International Conference on Advances in the Emerging Computing Technologies (AECT), pp. 1–6. IEEE (2019).

[11] Rogge-Solti, A., Kasneci, G.: Temporal Anomaly Detection in Business Processes. In: Sadiq, S., Soffer, P., Völzer, H. (eds.) Business Process Management. LNCS, vol. 8659, pp. 218–233. Springer, Cham (2014).

[12] J. L. Pereira and J. Varajão, "The Temporal Dimension of Business Processes - Dealing with Time Constraints," Procedia Computer Science, vol. 121, pp. 1034-1038, 2017.

[13] Senderovich, A., Weidlich, M., Gal, A. Temporal Network Representation of Event Logs for Improved Performance Modelling in Business Processes. In: Carmona, J., Engels, G., Kumar, A. (eds) Business Process Management. BPM 2017. Lecture Notes in Computer Science, vol 10445. Springer, Cham. (2017).

[14] Böhmer, K., & Rinderle-Ma, S. Multi instance anomaly detection in business process executions. In Business Process Management: 15th International Conference, BPM 2017, Barcelona, Spain, September 10–15, 2017, Proceedings 15 (pp. 77-93). Springer International Publishing. (2017)

[15] Janina, Sontheim., Florian, Richter., Thomas, Seidl. (2018). Temporal Deviations on Event Sequences.

[16] Florian Richter, Thomas Seidl, Looking into the TESSERACT: Time-drifts in event streams using series of evolving rolling averages of completion times, Information Systems, Volume 84, 2019, Pages 265-282, ISSN 0306-4379.

[17] Mavroudopoulos, I., & Gounaris, A. (2020). Detecting temporal anomalies in business processes using distance-based methods. In Discovery Science: 23rd International Conference, DS 2020, Thessaloniki, Greece, October 19–21, 2020, Proceedings 23 (pp. 615-629). Springer International Publishing.

[18] D. Zhao and Z. Zhou, "Anomaly and Interestingness Detection in Timed Hierarchical Business Processes," in IEEE Transactions on Engineering Management.

[19] Richter, F., Lu, Y., Zellner, L., Sontheim, J., & Seidl, T. (2020, October). TOAD: trace ordering for anomaly detection. In 2020 2nd International Conference on Process Mining (ICPM) (pp. 169-176). IEEE.

[20] Asma Ouarhim, Karim Baina and Brahim Elbhiri, "Towards a Continuous Temporal Improvement Approach for Real-Time Business Processes" International Journal of Advanced Computer Science and Applications(IJACSA), 15(1), 2024.

[21] Qafari, M.S., van der Aalst, W. Root Cause Analysis in Process Mining Using Structural Equation Models. In: Del Río Ortega, A., Leopold, H., Santoro, F.M. (eds) Business Process Management Workshops. BPM 2020. Lecture Notes in Business Information Processing, vol 397. (2020) Springer, Cham.

[22] Luo, Qian, Zhang, Lin, Xing, Zhiwei, Xia, Huan, Chen, Zhao-Xin, Causal Discovery of Flight Service Process Based on Event Sequence, Journal of Advanced Transportation, 2021, 2869521, 17 pages, 2021.

[23] Van Houdt, G., Depaire, B., Martin, N. (2022). Root Cause Analysis in Process Mining with Probabilistic Temporal Logic. In: Munoz-Gama, J., Lu, X. (eds) Process Mining Workshops. ICPM 2021. Lecture Notes in Business Information Processing, vol 433. Springer, Cham.

[24] Qafari, M.S., van der Aalst, W.M.P. Feature recommendation for structural equation model discovery in process mining. Prog Artif Intell (2022)

[25] Van Der Aalst, W., & van der Aalst, W. (2016). Data science in action (pp. 3-23). Springer Berlin Heidelberg.

[26] Sarang, P. . Gaussian Mixture Model. In: Thinking Data Science. The Springer Series in Applied Machine Learning. (2023) Springer, Cham

[27] Dablander, F. (2020). An introduction to causal inference.

[28] Zanga, A., Ozkirimli, E., & Stella, F. (2022). A survey on causal discovery: theory and practice. International Journal of Approximate Reasoning, 151, 101-129.

[29] Chen, W., Cai, R., Zhang, K., & Hao, Z. (2021). Causal discovery in linear non-gaussian acyclic model with multiple latent confounders. IEEE Transactions on Neural Networks and Learning Systems, 33(7), 2816-2827.

[30] Nogueira, A. R., Pugnana, A., Ruggieri, S., Pedreschi, D., & Gama, J. (2022). Methods and tools for causal discovery and causal inference. Wiley interdisciplinary reviews: data mining and knowledge discovery, 12(2), e1449.

[31] Tharwat, A. (2021), "Independent component analysis: An introduction", Applied Computing and Informatics, Vol. 17 No. 2, pp. 222-249.

4.6 Temporal Heterogeneous Causal Effects in Organizational Mining via Multiview Autoencoders

Building on the structural, organizational, and causal monitoring results, the fifth article addresses the contextual heterogeneity of interventional effects. The study is situated at the interventional level of the hierarchy (**L_2**) and jointly considers the organizational perspective, temporal deviations, and the process variant (**SO.2**, **RQ.3**). Multiview representations and double machine learning techniques, including Orthogonal Random Forest (**ORF**), are used to obtain context-conditioned causal effects. At the domain level, the article proposes a multi-perspective causal process mining scheme oriented towards the estimation of heterogeneous causal effects.

Publication Summary

Title:	*Temporal Heterogeneous Causal Effects in Organizational Mining via Multiview Autoencoders*
Authors:	Fernando Montoya, Gonzalo Rivera-Lazo, Mauricio Hidalgo, Manuel Goyo and Hernán Astudillo
Conference:	IEEE Technology and Engineering Management
Publisher:	IEEE *Xplore*
Year:	2025
Doi:	https://doi.org/10.1109/TEMSCONLATAM65810.2025.11238525

Temporal Heterogeneous Causal Effects in Organizational Mining via Multiview Autoencoders

Fernando Montoya[1,2], Gonzalo Rivera-Lazo[3], Mauricio Hidalgo[4], Manuel Goyo[1], Hernán Astudillo[5]

[1]Universidad Técnica Federico Santa María, Santiago, Chile. {fernando.montoya, manuel.goyo}@usm.cl
[2]Fundación Instituto Profesional Duoc UC, Santiago, Chile
[3]Corriente Sur, Valparaíso, Chile. gonzalo@corrientesur.com
[4]Universidad Finis Terrae, Providencia, Chile. mhidalgo@uft.cl
[5]ITiSB - Universidad Andrés Bello, Viña del Mar, Chile. hernan.astudillo@unab.cl

***Abstract*—**
Temporal variability in business processes represents a significant challenge in organizational mining, driven by changes in demand, regulatory requirements, or internal restructuring. These factors reconfigure the causal relationships among process roles, altering their interaction patterns. Detecting such changes in a timely manner is crucial to maintaining operational efficiency. Analyzing roles in temporally dynamic processes requires incorporating multiple perspectives to capture complex patterns in their evolving interactions. Interventions applied to these roles yield heterogeneous effects depending on their groupings and temporal contexts. To address this complexity, we propose an approach based on multiview autoencoders that capture diverse interaction perspectives among roles and enable clustering based on their latent patterns. This methodology uncovers specific causal relationships and estimates heterogeneous effects, supporting adaptive interventions for the roles involved. Validation was carried out using data from a payment services company, which revealed clusters of roles with similar patterns and identified their distinct causal relationships between temporal scenarios. This insight guided the design of targeted and focused interventions. This work contributes to organizational mining in temporally variable contexts through the integration of multiview autoencoders and causal inference techniques, thereby enabling adaptive interventions aligned with role-interaction patterns and their operational context.

***Index Terms*—organizational mining, business process mining, causal inference, multiview encoding.**

I. Introduction

Organizational mining is a valuable tool for analyzing event logs that reveal how business processes are executed. It helps in understanding resource allocation and interaction among roles [1, 2]. In highly interdependent environments, the temporal variability in execution times is crucial. Fluctuations can delay activities, create bottlenecks, and affect operational efficiency [3, 5]. This is observed in sectors such as banking, manufacturing, or software development, where delays may lead to cost overruns, regulatory noncompliance, or critical failures.

Several studies on process integration and organizational networks have highlighted the importance of the organizational perspective. For example, [17] proposes methods for identifying organizational structures from event logs. Focuses on how actors are grouped according to shared activities or cases. In contrast, [18] analyzes how work is transferred between individuals. Its approach allows us to observe dynamic interactions within the process.

However, both studies assume static structures. They do not consider temporal variability, such as changes in the duration of activities. Nor do they explore underlying and heterogeneous relationships that affect such variability. Causal inference techniques have shown the ability to estimate heterogeneous effects [7, 8]. This suggests that they can be adapted to identify hidden relationships that influence the temporal dynamics of processes.

Figure.1 shows the importance of temporal variability and characterizes the problem. At the *Detailed Trace* level, different process traces are observed as $\sigma = \langle e_1, e_2, \ldots, e_n \rangle$, where each event e_i corresponds to the execution of an activity (*for example*, A, B, C, D, E, F) carried out by different roles. Variations in execution times, highlighted in the figure, can alter the sequence and affect coordination among actors [3]. At the *Pattern Trace* level, execution patterns are abstracted, revealing how certain combinations of activities can change in order or duration and consequently influence the overall performance of the process [4].

This phenomenon is critical in contexts such as financial approval, where a delay in review (activity B) affects subsequent activities (C and D), or in software development, where late delivery of a module affects integration and final testing. Thus, the central challenge is to understand and model temporal variability and its underlying relationships in organizational processes. The problem is that traditional methods do not integrate multiple organizational perspectives (*for example, execution times, roles, and resources*) into a unified representation, nor do they dynamically explain the underlying causal relationships behind delays. As a result, there is no framework that identifies and quantifies the heterogeneous effects stemming from variations in activity duration [5, 6].

Based on these considerations, the following research questions are posed:

979-8-3315-2567-5/25/$31.00

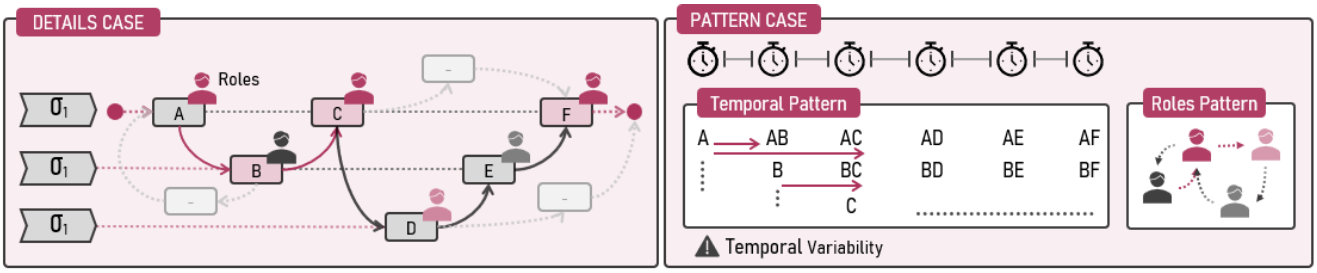

Fig. 1: Example of temporal variability from two perspectives. In the *Detailed Case*, the trace $\sigma = \langle e_1, e_2, \ldots, e_n \rangle$ is observed with execution times that vary by role. In the *Pattern Case*, these variations are abstracted, showing how changes in order or duration can impact overall efficiency.

- RQ_1: How does temporal variability manifest itself in highly interdependent scenarios and how does it impact operational efficiency?
- RQ_2: What methods enable the identification and modeling of the causal relationships underlying delays in key activities?

The document is organized as follows: Section II, Preliminaries, introduces key concepts; Section III, Related work, reviews previous studies; and Section IV, Proposed Methodology, describes the approach based on process mining and causal inference. Subsequently, Section V presents the results, followed by Discussion and Conclusions in Section VI.

II. Preliminaries

This section presents the formal elements necessary to establish the conceptual framework of the study. Its purpose is to clarify the notation and fundamental assumptions on which the subsequent analysis is built.

Definition 1 (Organizational Perspective). *Let L be an event log. Each trace $\sigma \in L$ is represented as an ordered sequence of events: $\sigma = \langle e_1, e_2, \ldots, e_n \rangle$. An event e_i is modeled as a tuple $(a_j, r_j, t_j^{start}, t_j^{end})$, where $a_j \in \mathcal{A}$ is the executed activity, $r_j \in \mathcal{R}$ is the assigned resource, and $t_j^{start}, t_j^{end} \in \mathbb{R}^+$ represent the start and end times, respectively. The duration of the event is defined as $d_j = t_j^{end} - t_j^{start}$. The organizational perspective focuses on the analysis of the function $r : e_j \mapsto r_j$, which links activities with resources, allowing the modeling of structures such as roles, teams, or functional units.*

Definition 2 (Temporal Variability). *Let $\sigma_i = \langle e_1^i, e_2^i, \ldots, e_n^i \rangle$ be a trace from the log L, where each event e_j^i has a timestamp $t_j^i \in \mathbb{R}^+$. The interval between two consecutive events e_j^i and e_{j+1}^i is defined as $\Delta t_{j,j+1}^i = t_{j+1}^i - t_j^i$. For a pair of activities $(a, b) \in \mathcal{A} \times \mathcal{A}$, the set of observed intervals is defined as $\mathcal{T}_{a,b} = \{\Delta t_{j,j+1}^i \mid a_j^i = a,\ a_{j+1}^i = b\}$. The average duration between a and b is denoted as $\overline{\Delta t}_{a,b} = \frac{1}{|\mathcal{T}_{a,b}|} \sum \mathcal{T}_{a,b}$, and the temporal deviation for a given case i is expressed as $\gamma_{a,b}^i = \Delta t_{j,j+1}^i - \overline{\Delta t}_{a,b}$. This value reflects the degree of deviation from the average temporal behavior observed between activities a and b.*

Definition 3 (Causal Graph). *Let $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ be a directed acyclic graph, where each node $X_j \in \mathcal{V}$ corresponds to a random variable and each directed edge $(X_i \to X_j) \in \mathcal{E}$ indicates that X_i exerts a direct causal influence on X_j. This structure reflects the dependency relationships generated by a causal model and defines a factorization of the joint distribution as $p(x_1, \ldots, x_d) = \prod_{j=1}^{d} f_j(x_j \mid x_{\mathrm{PA}_j})$, where $\mathrm{PA}_j \subseteq \mathcal{V}$ is the set of parents of X_j in the graph $\mathcal{G}$.*

Definition 4 (Causal Effect). *Let $X \in \mathbb{R}^p$ be the input variable and $Y \in \mathbb{R}$ the output. X is said to have a causal effect on Y when an intervention $do(X = x)$ changes the distribution of Y. The effect of changing X from x' to x is defined as $\tau(x, x') = \mathbb{E}[Y(x)] - \mathbb{E}[Y(x')]$, where $Y(x)$ is the potential outcome under that intervention.*

Definition 5 (Heterogeneous Causal Effects). *Let $T \in \{0, 1\}$ be a binary indicator (1 = treatment, 0 = control). For each unit, the potential outcomes are defined as $Y(1)$ (if the unit receives the treatment) and $Y(0)$ (if not). Heterogeneity exists when the individual effect $\tau_i = Y_i(1) - Y_i(0)$ varies across units; to describe it, the CATE is used: $\tau(x) = \mathbb{E}[\tau_i \mid X = x]$, which quantifies the expected treatment effect for units with covariates $X = x$, thus avoiding the limitations of a global average.*

III. Related Work

This section reviews advances in organizational mining from three main perspectives: modeling temporal variability, integrating multiview perspectives, and expanding data sources. The aim is to situate the current state of the field and to highlight the limitations that motivate this work.

A. Methods and Approaches for Temporal Variability in Organizational Mining

The analysis of organizational structures has evolved from static approaches to models more sensitive to execution dynamics. Appice et al. propose detecting overlapping communities in event flows, allowing the representation of organizational units that change over time [15]. Each event is considered part of a trace $\sigma = \langle e_1, e_2, \ldots \rangle$, where resources can participate in multiple communities. However, the model does not use timestamps t_j nor quantify execution intervals $\Delta t_{j,j+1}$.

Deokar and Tao introduce `OrgMiner`, a framework that detects interaction patterns between resources based on event logs [16]. The approach identifies activity sequences executed by different actors, but does not explicitly consider differences in execution times or their dispersion. In this context, measures such as $\gamma^i_{a,b}$ would be useful in characterizing temporal variability.

Gao et al. address the problem through similarity metrics between resources, based on how frequently they participate in similar activities or cases [17]. These relationships can be represented by a function $w(r_i, r_j)$, which assigns a weight to each pair of resources according to their co-occurrence. Although the model captures recurrent patterns, it does not consider execution time or associated variability.

Syamsiyah et al. propose moving processing to the database level, allowing organizational relationships to be efficiently precomputed [18]. Each event is stored with attributes such as *activity* (a_j), *resource* (r_j), *timestamp* (t_j), enabling formal structuring. However, the derived relationships remain static and do not reflect temporal changes or causal structures.

B. Multiview Perspectives in Organizational Processes

Incorporating multiple views allows for a better representation of the complexity of collaborative processes. Some studies combine information on activities, resources, time, and relationships, although with varying levels of integration.

Wang et al. combine process mining and social network analysis to model collaborative supply chains [19]. Their approach enables the analysis of events that represent work transfers between actors, with temporal information such as t_j and sequences (e_j, e_{j+1}). However, they do not quantify the intervals $\Delta t_{j,j+1}$ or assess their variability.

Zeng et al. use extended Petri nets to represent interorganizational flows with activity synchronization, messaging, and shared resources [20]. Their model captures complex structural relationships, but does not incorporate time measurements or evaluate how these relationships change across different executions.

Qiu et al. introduce the concept of a *Community Tree* to model hierarchical structures that evolve over time [21]. Their approach identifies the most influential actors within a social network and organizes them into hierarchical trees. Each member is associated with an immediate superior based on their level of importance. By comparing these structures over time, changes in organizational configuration are represented. Although the model captures structural dynamics, it is not linked to temporal measures or articulated with other process dimensions.

Together, these studies show that the integration of multiple viewpoints is possible, but the lack of connection between organizational, temporal, and causal aspects limits their explanatory power.

C. Expansion of Data Sources in Organizational Mining

Abdelkafi et al. use natural language processing to analyze emails and extract implicit organizational relationships [22]. Based on attributes such as sender, recipient, and sending time t_j, they construct event logs that reflect unstructured interactions. Although these enriched traces broaden the analysis, they are not connected to other views and neither is the temporal variation between activities measured.

Engel et al. analyze `EDI` data to model interorganizational processes [23]. The messages are transformed into formally structured events, allowing the reconstruction of collaboration sequences between actors. While their approach allows for the evaluation of transactional performance, it does not incorporate temporal variability metrics nor analyze how these relationships evolve.

IV. Proposed Technique

In this section, a four-stage methodology transforms process logs into latent representations to identify patterns and causal relationships. Figure.2 summarizes the flow from event log extraction to cluster-based causal inference.

A. Feature Extraction From Business Processes

In `STAGE` 4.1, the business process to be analyzed is selected by identifying the sequence of activities that make up its execution variants. The event log is captured from systems such as `ERP` (*Sap, Oracle*), `BPM` (*Camunda, Bizagi*), or `CRM` (*Salesforce*), which are commonly used in organizational environments. This event log serves as a structured data source that is a key in process mining [24].

In `STAGE` 4.2, the log is represented as a collection of traces σ_i, where each event e_j contains attributes such as activity, time, and resource. Using the `XES` format [25], the log structure is standardized (*see* Scheme.1). From the traces, specific views are derived: time view, roles view, and flow view, which are used as input in the following stages.

Scheme of an event log in XES format

```
<log>
<trace>
<string key="concept:name" value="Case_12345"/>
<event>
  <string key="concept:name" value="Start Process"/>
  <string key="org:role" value="System"/>
  <date key="time:timestamp" value="2023-01-15 09:00:000"/>
  <string key="custom:op" value="init"/>
</event>
    <!-- ... additional events ... -->
<event>
  <string key="concept:name" value="Payment Verification"/>
  <string key="org:role" value="Financial_Analyst"/>
  <date key="time:timestamp" value="2023-01-15 09:45:00"/>
  <string key="custom:op" value="verify_payment"/>
  <string key="amount" value="1250.00"/>
</event>
    <!-- ... more events ... -->
</trace>
    <!-- ... additional traces ... -->
</log>
```

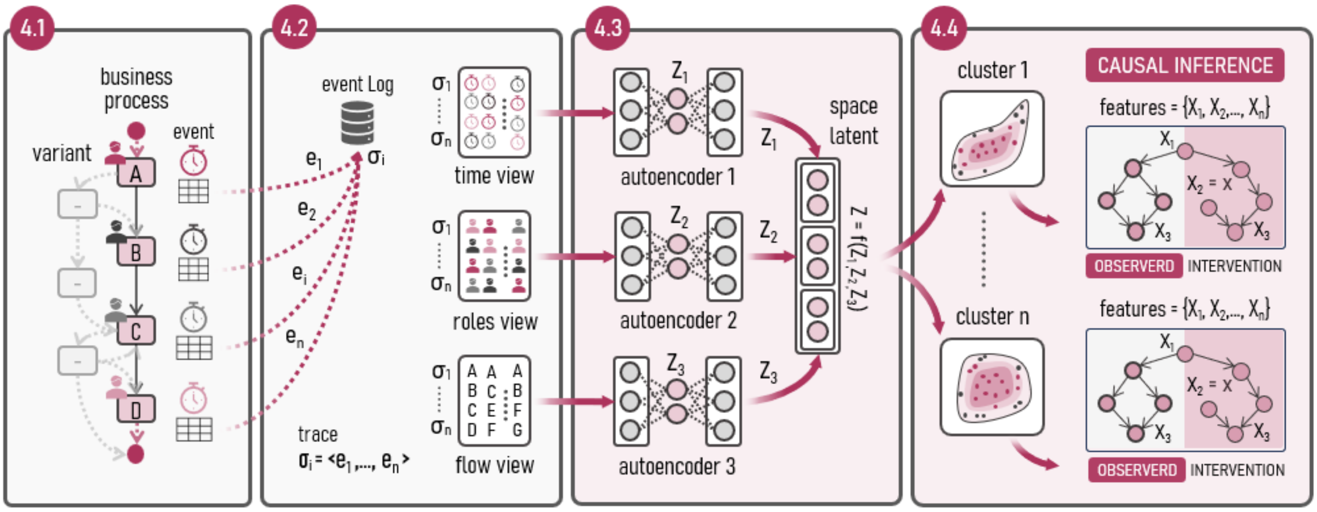

Fig. 2: Methodological approach integrating organizational and temporal views via multi-view autoencoders, transforming data into a combined latent space for clustering and causal inference.

B. Multiview Encoding with Autoencoders

STAGE 4.3 converts the obtained views into latent representations using independently trained auto-encoders. This encoding abstracts relevant patterns in each process dimension and provides a compact representation of each trace for subsequent analysis.

An *autoencoder* is a self-supervised neural network composed of an encoder $f_{\text{enc}} : \mathbb{R}^d \to \mathbb{R}^k$ and a decoder $f_{\text{dec}} : \mathbb{R}^k \to \mathbb{R}^d$, where $k < d$. The encoder projects the input $\mathbf{x}$ into a latent representation $\mathbf{z} = f_{\text{enc}}(\mathbf{x})$, approximating the conditional distribution $p(\mathbf{z} \mid \mathbf{x})$.

Subsequently, the *decoder* attempts to reconstruct the original input $\hat{\mathbf{x}} = f_{\text{dec}}(\mathbf{z})$, approximating the conditional distribution $p(\mathbf{x} \mid \mathbf{z})$. This mechanism forces the model to learn intrinsic data patterns.

A separate instance is trained for each process perspective: activities (*flow view*), roles (*roles view*), and execution time (*time view*). For each trace $\sigma_i = \langle e_1^i, e_2^i, \ldots, e_n^i \rangle$, the input vectors are defined as:

- ACTIVITIES: The vector $\mathbf{x}_f^i = [a_1^i, a_2^i, \ldots, a_n^i] \in \mathbb{R}^n$ is built, where each a_j^i is the integer value assigned to the jth activity of the trace using ordinal encoding.
- ROLES: Similarly, $\mathbf{x}_r^i = [r_1^i, r_2^i, \ldots, r_n^i] \in \mathbb{R}^n$, where each r_j^i represents the encoded role of the event e_j^i.
- TIME: The vector $\mathbf{x}_t^i = [\delta_{a,b}^i] \in \mathbb{R}^m$ (*see* def.2) is computed, composed of values transformed using the modified z-score. Each component $\delta_{a,b}^i$ represents the normalized temporal deviation for the activity pair (a, b), based on the interval $\Delta t_{j,j+1}^i = t_{j+1}^i - t_j^i$.
  From the previously defined set $\mathcal{T}_{a,b}$, the following is calculated: $\text{MAD}_{a,b} = \text{median}\left(|\Delta t - \text{Med}_{a,b}| \mid \Delta t \in \mathcal{T}_{a,b}\right)$ and $\text{Med}_{a,b} = \text{median}(\mathcal{T}_{a,b})$

$$\delta_{a,b}^i = 0.6745 \cdot \frac{\Delta t_{j,j+1}^i - \text{Med}_{a,b}}{\text{MAD}_{a,b}}$$

Therefore, the input vector $\mathbf{x}_t^i$ is constructed by considering all ordered pairs of activities in the trace σ_i, preserving their order of occurrence. Each component of this vector represents the temporal variation $\delta_{(a,b)}^i$ calculated for each pair (a, b) according to the previously defined metric. Formally, the temporal input vector is $\mathbf{x}_t^i = [\delta_{(a_j^i, a_k^i)}^i \mid 1 \le j < k \le n]$, with dimension $|\mathbf{x}_t^i| = \binom{n}{2}$. For example, given the trace $\sigma_i = [a, b, c, d]$, we obtain $\mathbf{x}_t^i = [\delta_{(a,b)}^i, \delta_{(a,c)}^i, \delta_{(a,d)}^i, \delta_{(b,c)}^i, \delta_{(b,d)}^i, \delta_{(c,d)}^i]$.

In this way, the relative temporal variations (*see* def.2) between all event pairs in the trace are captured. Finally, each view is encoded separately into latent representations $\mathbf{z}_f^i$, $\mathbf{z}_r^i$, and $\mathbf{z}_t^i$ [12], which are integrated by concatenation:

$$\mathbf{z}^i = \mathbf{z}_f^i \oplus \mathbf{z}_r^i \oplus \mathbf{z}_t^i.$$

C. Latent Clustering and Heterogeneous Causal Modeling

In STAGE 4.4, work is carried out on the set of latent representations $\mathcal{Z} = \{\mathbf{z}^i\}_{i=1}^n$, on which a clustering task is applied to segment the traces into subsets that share structural similarity. Each cluster C_k represents a region of the latent space associated with a recurrent pattern of organizational behavior.

For each cluster, the original traces $\{\sigma^i \mid \mathbf{z}^i \in C_k\}$ are recovered and the operational variables $x = \{x_1, \ldots, x_p\}$ are extracted, which are used to identify causal relationships (*see* def.4) through the LiNGAM [9], PC [10], and GES [11] algorithms (*see* alg.1). A directed edge between variables is considered significant if it appears predominantly in the results of any of the algorithms, thus establishing a partial consensus criterion on the cluster's causal structure.

From the identified edges, a pair of variables (T_k, Y_k) is selected, where T_k is considered the causal variable and Y_k the outcome variable (*see* def.4). In order to estimate how the effect of T_k on Y_k varies depending on the operational context, the orthogonal random forest (ORF) [13] is used, a

Algorithm 1 Bootstrapped Causal Graph Estimation

Input : $\mathbf{D} \in \mathbb{R}^{n \times p}$ (data matrix with p variables)
$B \in \mathbb{N}$ (number of bootstrap iterations)
$\tau \in [0, 1]$ (threshold for edge inclusion)
Output : $\mathcal{G}^* = (V, E^*)$, where $V = \{X_1, \dots, X_p\}$

$v_{ij} \leftarrow 0$ for all $(i, j) \in V \times V$ /* *Initialize edge counts* */
for $b \leftarrow 1$ **to** B **do**
  $\mathbf{D}^{(b)} \leftarrow$ bootstrap sample from $\mathbf{D}$ **for each** $(i, j) \in V \times V$ **do**
    if $X_i \rightarrow X_j$ *detected in* $\mathbf{D}^{(b)}$ *by any* $m \in \{LiNGAM, PC, GES\}$ **then**
      $v_{ij} \leftarrow v_{ij} + 1$ /* *Increment edge count* */
$f_{ij} \leftarrow \frac{v_{ij}}{B}$ for all $(i, j) \in V \times V$ /* *Compute edge frequencies* */
$E_{\text{init}} \leftarrow \{(i, j) \mid f_{ij} \geq \tau\}$ /* *Select significant edges* */
$\mathcal{G}_{\text{init}} \leftarrow (V, E_{\text{init}})$
while $\mathcal{G}_{init}$ *contains a cycle* $\mathcal{C}$ **do**
  /* *Find separable edges* */
  $E_{\text{sep}} \leftarrow \{(i, j) \in \mathcal{C} \mid \text{d-sep}(X_i, X_j, \mathbf{Z}) = \text{True}\}$
  if $E_{sep} \neq \emptyset$ **then**
    $(i' j') \leftarrow_{(i,j) \in E_{\text{sep}}} v_{ij}$ /* *Lowest confidence edge* */
    $E_{\text{init}} \leftarrow E_{\text{init}} \setminus \{(i' j')\}$ /* *Remove edge* */
    Update $\mathcal{G}_{\text{init}}$
  else
    break /* *Cannot resolve cycle* */
return $\mathcal{G}^* \leftarrow \mathcal{G}_{init}$ /* *Return final acyclic graph* */

technique from the field of causal machine learning aimed at estimating conditional causal effects. The model operates on observations (X_i, T_i, Y_i) extracted from the cluster and estimates the function:

$$\tau_k(x) = \mathbb{E}[Y_k \mid T_k = 1, X = x] - \mathbb{E}[Y_k \mid T_k = 0, X = x],$$

where x represents the contextual conditions observed in the trace. ORF makes it possible to capture the variation of the causal effect according to the environment without requiring an explicit functional form or assuming the homogeneity of the impact across cases (*see* def.5). In this way, differences in the causal response within the cluster are identified, revealing heterogeneous effects structurally associated with specific execution conditions.

V. Results

A. Dataset

The dataset covers a real organizational process for embossing and personalizing payment cards. Each trace σ_i denotes a sequence of events with activity, timestamp, and assigned role (*see* def.1 and def.2), which the mining process [24] uses to accurately reconstruct the process performed in detail.

Figure.3 shows the process model[1] from the control flow perspective. All instances initiate at RECEIVE APPLICATION and conclude at one of two endpoints SHIPPING or CASE CLOSURE spanning a total of 23 distinct activities. The diagram reveals varied execution paths, including loops, repetitions, and alternative routes, which highlight the presence of process variants as valid but different sequences of activities within this single model. The event log comprises 7,526 traces. Each trace begins with the same initial event; approximately 67 % terminate at one endpoint, while the remaining 33 % reach the other. This dataset spans a continuous 30-day interval, providing context for analyzing variant evolution. (*see* Table.I).

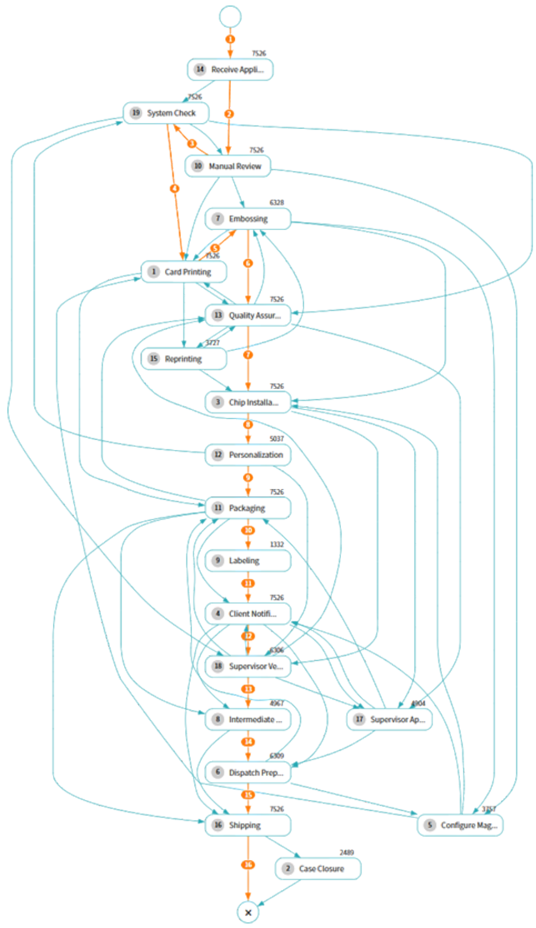

Fig. 3: Process flow model for card personalization that illustrates multiple execution paths, including loops and alternatives, within the personalization process.

TABLE I: Summary of process characteristics

Attribute	Value
Number of recorded traces	7,526
Number of activities	19
Number of roles	12
Dominant initial activity	Receive Application (100%)
Main final activities	Shipping (67%), Case Closure (33%)
Linear flow	No (branched flow)
Role connectivity	High
Time span of the log	30 days

[1]The process model was generated using ABBYY Timeline – Process Mining platform. https://www.abbyy.com/timeline/

B. Latent Representation and Segmentation

From the multiview encoding described in the methodological section (*see* Subsect.4.3), a fused latent space was generated by combining the representations obtained through independent autoencoders for activity, role, and time views. This space was used to segment the traces according to integrated structural and contextual patterns.

The SPECTRAL CLUSTERING technique [14], chosen for its ability to capture non-linear cluster structures without assuming Gaussian distributions, unlike probabilistic methods such as GMM, was applied to this latent space, resulting in six differentiated clusters. The number of clusters was set to six based on preliminary validation using silhouette and Davies–Bouldin indices. Figure.4 illustrates the distribution of the 7,526 traces projected in two dimensions, showing a clear separation among the generated groups.

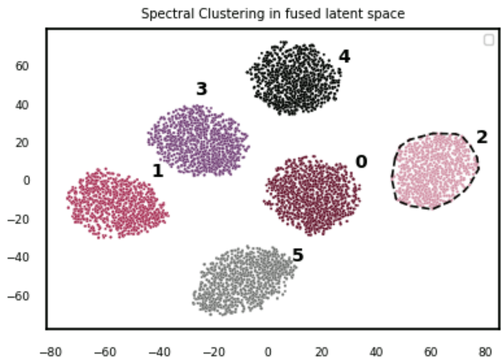

Fig. 4: SPECTRAL CLUSTERING applied to the fused latent space.

Cluster 2 was selected for causal analysis due to its greater dispersion, which indicated variability in trace structure and increased the likelihood of identifying diverse interaction patterns and context-dependent effects. To measure cluster cohesion and internal dispersion, inertia and average distance were calculated. Table.II presents these values. Cluster 2 showed the highest inertia (62,960.63) and the highest average distance (7.05)

TABLE II: Summary of internal metrics by cluster

Cluster	Size	Inertia	Avg. Dist.
0	1332	62007.74	6.72
1	1217	22959.22	4.26
2	**1220**	**62960.63**	**7.05**
3	1290	52846.03	6.30
4	1198	58378.51	6.85
5	1269	45439.21	5.86

Subsequently, the role collaboration network associated with the traces of this cluster was examined. Figure.5 shows the interactions between the process participants, revealing a collaboration pattern centered on three activities: INTERMEDIATE STORAGE (role: *Logistics Staff*) and DISPATCH PREPARATION (role: *Shipping Coordinator*). These transitions indicate a key operational flow within the cluster. The

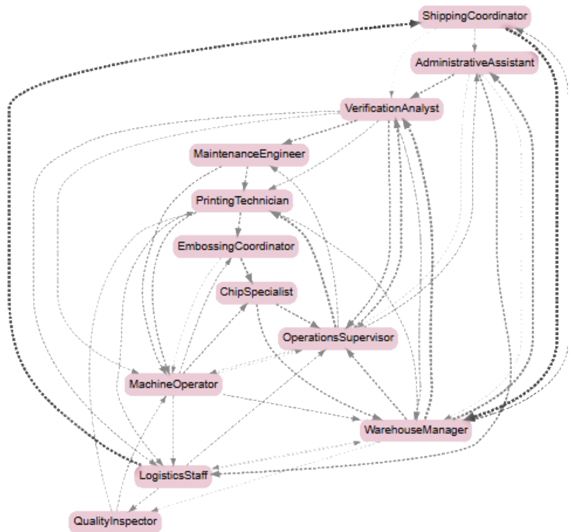

Fig. 5: Role-collaboration network for cluster 2, derived from the fused latent space.

operational variables of these activities, case volume, rejection rate, and relative duration, were selected to estimate conditional effects according to the contexts of each cluster.

C. Causal Analysis and Heterogeneous Effects

Initially, the analysis focused on the activity INTERMEDIATE STORAGE within cluster C_2. The causal graph (Figure.6)(*see* def.3) revealed direct relationships from *Storage Temperature* and *Humidity Level* to *Damage Incidence*, and from the latter to *Security Alert Level*, detected by GES (*green arc*) [11] and partially by PC + GES (*purple arc*) [10].

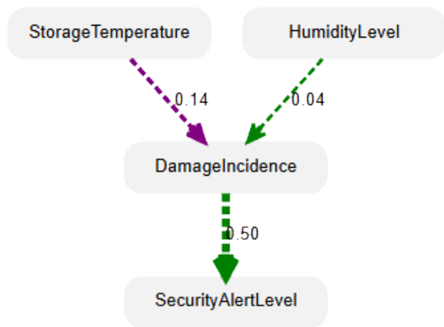

Fig. 6: Estimated causal graph for INTERMEDIATE STORAGE.

Storage Temperature was set as the treatment and *Security Alert Level* as the outcome; *Humidity Level* and baseline *Damage Incidence* were used as effect modifiers to capture operator variability. Figure.7 shows that, at humidity levels between 47.1 % and 47.3 %, lowering the temperature reduces the Security Alert Level by –0.30/–0.24 units (negative values indicate a reduction); with damage incidence of 0.19–0.20, the decrease is –0.27/–0.26 units. The benefit is concentrated at 40–60 % humidity (dispersion 0.16 between 50–56 %) and fluctuates ±0.18 units when damage ranges from –0.66 to 0.84.

Two critical scenarios stand out: 47 runs with mean humidity 53.5 % and damage 0.84 show a positive mean effect (+0.13 units), indicating that cooling under very humid conditions can increase alerts; in contrast, 18 runs with 46 % humidity and 0.17 damage show a protective effect of –0.58 units[1], indicating that thermal adjustment reduces alerts only under relatively dry initial conditions.

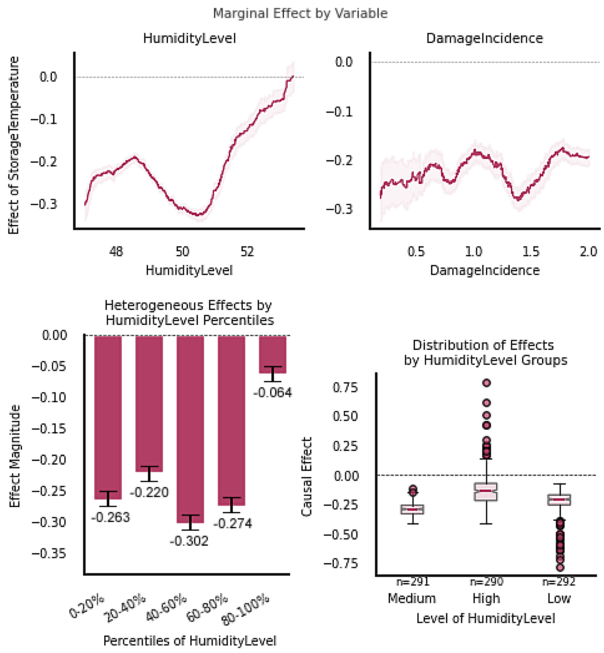

Fig. 7: Heterogeneous effects for `INTERMEDIATE STORAGE`.

Following the previous analysis, a second scenario was evaluated within the cluster C_2, corresponding to the activity `DISPATCH PREPARATION`. The estimated causal graph is presented in Figure.8. It shows that *Transit Time* and *Customer Satisfaction* directly determine *Shipping Cost*, which in turn influences *Delivery Success Rate*; detected by `GES` (*green arc*) [11] and partially by `PC` (*red arc*) [10].

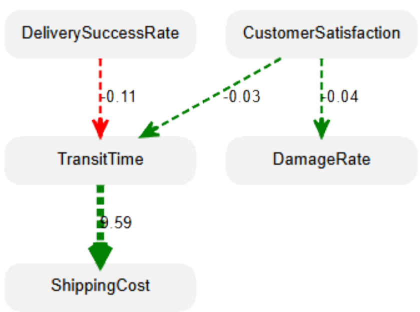

Fig. 8: Estimated causal graph for `DISPATCH PREPARATION`.

Delivery Success Rate was defined as the treatment and *Shipping Cost* as the outcome; *Customer Satisfaction* and baseline *Transit Time* were included only to reveal effect variation. Despite an overall mean effect near zero (0.0007 ± 0.0057), Figure 9 reveals pronounced heterogeneity: when transit time is around 11.85 h, cost increases to 0.006 units, but once transit exceeds 13 h, the effect turns negative (–0.0025 units). Customer satisfaction exhibits a similar pattern: cost rises near scores of 78–79, reaches a minimum of –0.005 units around 82, and climbs to 0.006 units when satisfaction exceeds 84. This trend holds across percentiles: the first two quintiles show cost increases close to 0.004, while the 60–80 % range concentrates the largest reduction (–0.005 units).

The subgroup analysis shows that 48 observations (mean satisfaction 85.2) incurred a cost increase of +0.012, while 74 observations with transit > 13 h saw a savings of -0.011. Thus, increasing delivery success only reduces costs for delayed shipments; in high satisfaction contexts, it increases service costs.

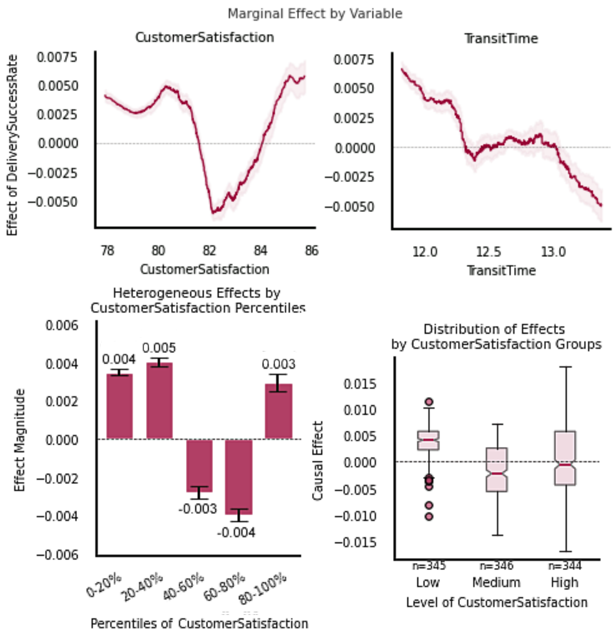

Fig. 9: Heterogeneous effects for `DISPATCH PREPARATION`

VI. Discussion & Conclusion

The results show that integrating the views of flow, roles, and time enables the segmentation of the traces into groups reflecting different operational conditions, allowing the identification of specific causal relationships within each segment. This form of analysis is particularly useful for processes with dynamic components, where execution conditions frequently change.

In contrast to other approaches that represent the organizational structure as fixed or summarized, such as those proposed in [15], [6], and [21], the methodology proposed here explicitly incorporates the temporal dimension. This makes it possible to estimate the effects conditioned by the operational context and to detect how factors such

[1]The "Low" group median is –0.25; these extreme values lower the mean.

as humidity, incidence of damage, or customer satisfaction influence the intensity and direction of causal effects. These nuances are difficult to capture through single-view methods. Additionally, combining different algorithms for causal discovery yields broader and more stable causal graphs, reducing reliance on a single method and enhancing the coverage of the generated model.

This approach directly addresses the research questions RQ.1 and RQ.2, demonstrating how temporal variability affects execution in interdependent contexts and providing a formal mechanism to identify and model causal relationships in key process activities.

The integration of multiview encoding, latent segmentation, and conditional causal analysis helps identify differentiated behavioral patterns and supports operational decision making with context-specific evidence.

A notable aspect is that this approach is not restricted to a particular domain, which makes it applicable to other organizational contexts characterized by high operational variability and complex interaction structures.

This adaptability is especially beneficial for processes designed to function in a standardized manner but exhibit significant execution differences in practice. The ability to decompose and model these behaviors using multiview logic and causal relationships provides a robust alternative to improve process monitoring, diagnosis, and adjustment in dynamic environments.

References

[1] Kirchmer, M., Gutiérrez, F., & Laengle, S. (2010). Process mining for organizational agility. Industrial Management.

[2] Zhao, W., & Zhao, X. (2014). Process Mining from the Organizational Perspective. In: Wen, Z., Li, T. (eds) Foundations of Intelligent Systems. Advances in Intelligent Systems and Computing. Springer.

[3] Song, M., & van der Aalst, W. M. P. (2008). Towards comprehensive support for organizational mining. Decision Support Systems.

[4] Schönig, S., Cabanillas, C., Jablonski, S., & Mendling, J. (2016). A framework for efficiently mining the organisational perspective of business processes. Decision Support Systems.

[5] Ahn, H., & Kim, K. P. (2020). Formal approach for discovering work transference networks from workflow logs. Information Sciences.

[6] Appice, A. (2018). Towards mining the organizational structure of a dynamic event scenario. Journal of Intelligent Information Systems.

[7] Huber, M. (2024). An introduction to causal discovery. Swiss Journal of Economics and Statistics.

[8] Salditt, M., Eckes, T., & Nestler, S. (2024). A Tutorial Introduction to Heterogeneous Treatment Effect Estimation with Meta-learners. Administration and Policy in Mental Health and Mental Health Services Research.

[9] Shimizu, S., Hoyer, P. O., Hyvärinen, A., Kerminen, A., & Jordan, M. (2006). A linear non-Gaussian acyclic model for causal discovery. Journal of Machine Learning Research.

[10] Spirtes, P., Glymour, C. N., Scheines, R., & Heckerman, D. (2000). Causation, prediction, and search. MIT press.

[11] Chickering, D. M. (2002). Optimal structure identification with greedy search. Journal of machine learning research.

[12] Li, P., Pei, Y., & Li, J. (2023). A comprehensive survey on design and application of autoencoder in deep learning. Applied Soft Computing

[13] Oprescu, M., Syrgkanis, V., & Wu, Z. S. (2019, May). Orthogonal random forest for causal inference. In International Conference on Machine Learning. PMLR.

[14] Von Luxburg, U. (2007). A tutorial on spectral clustering. Statistics and computing.

[15] Appice, A., Malerba, D., & Di Pietro, M. (2020). Discovering and tracking dynamic resource communities in business event streams. Journal of Intelligent Information Systems

[16] Amit, V., Deokar, & Jie, Tao. (2021). OrgMiner: A Framework for Discovering User-Related Process Intelligence from Event Logs. Information Systems Frontiers.

[17] Gao, A., Yang, Y., Zeng, M., Zhang, J. -L., & Wang, Y. -W. (2009). Organizational Structure Mining Based on Workflow Logs. 2009 International Conference on Business Intelligence and Financial Engineering, China.

[18] Syamsiyah, A., van Dongen, B. F., & van der Aalst, W. M. (2017). Discovering social networks instantly: Moving process mining computations to the database and data entry time. In BPMDS/EMMSAD 2017.

[19] Wang, J., Zhu, B., Wang, Y., & Huang, L. (2016). Mining organizational behaviors in collaborative logistics chain: An empirical study in a port. 2016 International Conference on Logistics, Informatics and Service Sciences (LISS), Sydney, NSW, Australia.

[20] Abdelakfi, M., Mbarek, N., & Bouzguenda, L. (2021). Mining organizational structures from email logs: An NLP based approach. Procedia Computer Science.

[21] Qiu, J., Lin, Z., Tang, C., & Qiao, S. (2009). Discovering Organizational Structure in Dynamic Social Network. 2009 Ninth IEEE International Conference on Data Mining, Miami Beach, FL, USA.

[22] Zeng, Q., Sun, S. X., Duan, H., Liu, C., & Wang, H. (2013). Cross-organizational collaborative workflow mining from a multi-source log. Decision Support Systems.

[23] Engel, R., Krathu, W., Zapletal, M., Pichler, C., Bose, R. J. C., van der Aalst, W., … Huemer, C. (2016). Analyzing inter-organizational business processes: Process mining and business performance analysis using electronic data interchange messages. Information Systems and e-Business Management

[24] Ahmed, R., Faizan, M., Burney, A.I.: Process Mining in Data Science: A Literature Review. In: 2019 International Conference on Advances in the Emerging Computing Technologies

[25] van der Aalst, W.M.P. (2022). Process Mining: A 360 Degree Overview. In: van der Aalst, W.M.P., Carmona, J. (eds) Process Mining Handbook. Springer

4.7 Causal Process Mining: From Interpretability to Causal Explainability in Temporal Deviations

The final article distinguishes between structural interpretability and causal explainability in temporal deviations. Interpretability is associated with the interventional level (**L_2**) and focuses on understanding the topology of the causal graph by variant. Explainability is linked to the counterfactual level (**L_3**) and is oriented towards answering, in a traceable way, "what would have happened if…" questions in decisions about process times (**SO.1**, **SO.2**; **RQ.2**, **RQ.3**). By integrating both levels in a single analysis, the work presents a causal process mining approach. This approach links structural identification with the counterfactual interpretation of interventions. In the literature, the result strengthens the connection between process mining and causal models by offering an explicit way to translate causal graphs into counterfactual explanations about temporal performance.

Publication Summary

Title: *Causal Process Mining: From Interpretability to Causal Explainability in Temporal Deviations*

Authors: Fernando Montoya, Mauricio Hidalgo and Hernán Astudillo

Conference: 8th International Conference on Systems Engineering (CIIS)

Publisher: ACM Digital Library

Year: 2025

Doi: https://doi.org/10.1145/3771678.3771681

Causal Process Mining: From Interpretability to Causal Explainability in Temporal Deviations

Fernando Montoya
Universidad Técnica Federico Santa María
Santiago, Chile
Fundación Instituto Profesional Duoc UC
Santiago, Chile
fernando.montoya@usm.cl

Mauricio Hidalgo
Universidad San Sebastián
Santiago, Chile
mauricio.hidalgob@uss.cl

Hernán Astudillo
ITiSB – Universidad Andrés Bello
Viña del Mar, Chile
hernan.astudillo@unab.cl

Abstract

In operations with timing constraints, drifts in execution times raise costs, weaken SLAs, and degrade user experience. While process mining reveals where and when deviations appear, correlation-based analyses rarely explain why they happen or what to change. We present a causal process mining method that moves from interpretability to causal explainability. First, we localize temporal deviations per process variant using performance spectra and cluster traces with similar interval profiles; then we learn a consensus causal graph (LiNGAM, PC, and GES) constrained by domain knowledge to make relationships readable and auditable. Second, on that graph we run intervention queries (*do-operator*) to answer "*what-if*" questions with identification conditions and uncertainty summaries, turning signals into actionable recommendations. We validate the approach on an event log derived from planning artifacts (1,261 traces; 6,015 events; 8 activities; 6 variants). The analysis uncovers causal routes that link complexity level and number of requirements to development/testing hours and meetings, and quantifies the expected impact of alternative policies on those outcomes. The method contributes: (*i*) variant-level causal graphs that preserve process heterogeneity; (*ii*) robust effect estimation that handles hidden heterogeneity across variants; (*iii*) spectrum-guided temporal localization of interventions; and (*iv*) explicit, auditable recommendations with confidence information to reduce temporal deviations.

CCS Concepts

• **Applied computing** → **Business process management**; • **Computing methodologies** → *Causal reasoning and diagnostics.*

Keywords

causal process mining, interpretability, causal explainability, temporal deviations, interventions, event logs

ACM Reference Format:
Fernando Montoya, Mauricio Hidalgo, and Hernán Astudillo. 2025. Causal Process Mining: From Interpretability to Causal Explainability in Temporal Deviations. In *8th International Conference on Systems Engineering - Cybersecurity & AI: Building a reliable digital future (CIIS 2025), October 01–03, 2025, Lima, Peru.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3771678.3771681

CIIS 2025, Lima, Peru

ACM ISBN 979-8-4007-1880-9/25/10
https://doi.org/10.1145/3771678.3771681

1 Introduction

Process mining has become a key mechanism for improving operations and guiding investments. In time-sensitive services (*e.g., healthcare, finance, logistics, public sector*), temporal deviations increase costs, affect service-level agreements, and degrade user experience. Although event logs allow observing actual process flows and their variability, translating these findings into actionable operational decisions remains challenging. Managing improvement initiatives requires explanations that connect observed phenomena with specific daily operational decisions [1].

In this work, the focus was on temporal deviations. Each trace $\sigma = \langle e_1, \ldots, e_m \rangle$ was represented by events of the form $e_k = (a_k, t_k, x_k)$, analyzing intervals $d_k = t_{k+1} - t_k$, where waiting times, queues, and bottlenecks typically emerge. Each trace was summarized with a robust score Y that aggregates these intervals. Recent literature describes patterns and declarative rules for identifying deviations but often relies on correlations vulnerable to confounders or mediators [2, 4, 5]. To overcome this, we adopted a causal approach, explicitly distinguishing temporal ordering from causal dependency, thus enabling operational decisions to be explained through explicit interventions.

We propose bridging analysis and practice in two stages: first, causal interpretability, obtaining a graph that identifies influential variables and their role in congestion propagation [3]; second, causal explainability, using the graph to assess actionable decisions and their operational impacts via interventions of the form $do(T^{(k)} = t)$, quantifying expected changes in deviations [6, 7]. Formally, the expected effect of these interventions was defined as:

$$\tau^{(k)}(t, t' \mid c) = \mathbb{E}_c\big[Y \mid do(T^{(k)} = t)\big] - \mathbb{E}_c\big[Y \mid do(T^{(k)} = t')\big] \quad (1)$$

Where $do(\cdot)$ indicates the causal intervention and c represents the observable context before the event. We then showed how to identify and estimate these causal impacts [6, 7]. Detailed temporal analysis allowed locating specific interventions. The Performance Spectrum revealed variability at the level of specific activities or segments, identifying queues, batching, and operational shifts [8, 9]. This served as a local diagnostic tool for selecting segments for intervention.

Finally, our work contributes: (***i***) extraction of causal graphs per variant using algorithms (LiNGAM, PC, and GES) and domain knowledge [3, 6, 7]; (***ii***) robust estimation of event-level causal effects [6, 7]; (***iii***) precise temporal localization using performance spectra [8]; and (***iv***) explicit causal recommendations, providing targeted interventions accompanied by confidence intervals and

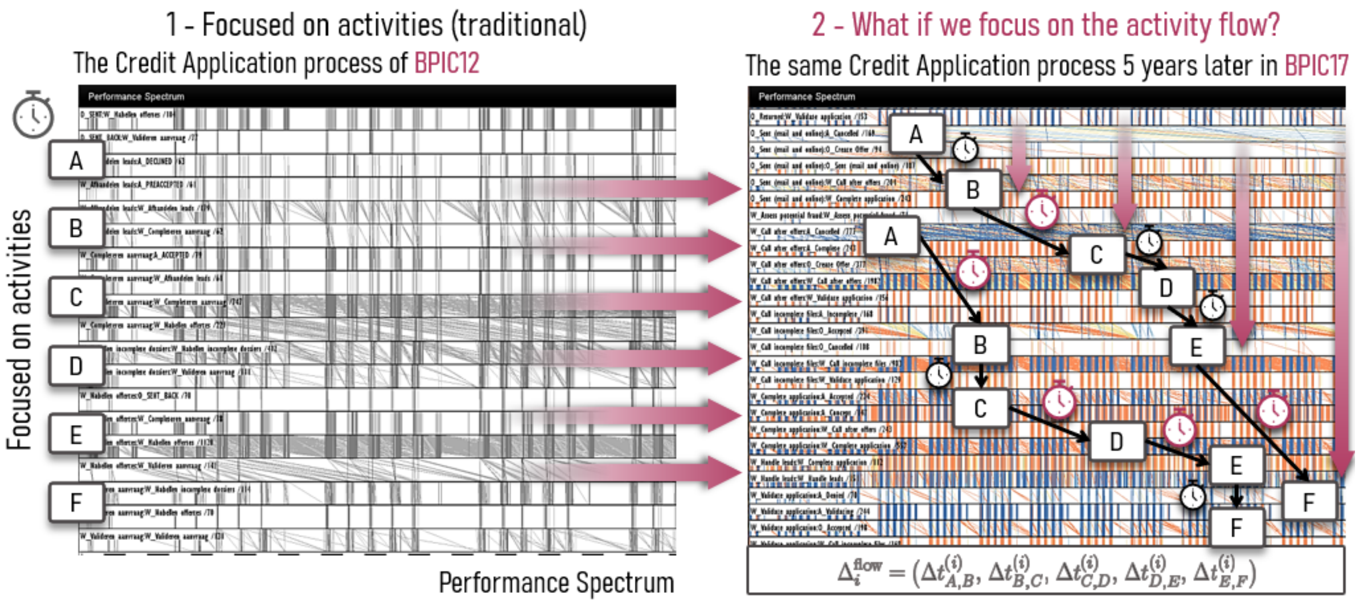

Figure 1: Process characterization: activity-based reading vs. flow-based reading. The left panel shows the temporal variability per activity (*traditional reading of the Performance Spectrum*); the right panel shows the temporal deviations along the flow (Δt^{flow}), highlighting the chaining of delays between transitions.

clearly stated assumptions [1]. To guide this work, we propose three questions: how can process mining be combined with causal techniques to explain temporal deviations, what causal relationships drive these deviations, and how such explanations improve operational decisions and interventions.

(1) ***RQ*1**. How can process mining be integrated with causal machine learning techniques to identify specific causes of temporal deviations?
(2) ***RQ*2**. What are the key causal relationships associated with the temporal deviations observed in business processes?
(3) ***RQ*3**. To what extent does causal understanding improve the quality and effectiveness of managerial decisions and interventions in response to temporal deviations?

The document is organized as follows: Section 2 reviews related work; and Section 3 describes the proposed methodology integrating process mining and causal inference. Next, Section 4 presents and analyzes the main results, followed by the Discussion in Section 5, and finally the Conclusions in Section 6.

2 Related Work

2.1 Temporal Deviations in Business Processes

In BPM, the focus shifted from structural to temporal aspects to understand deviations and support decisions. Rogge-Solti and Kasneci applied a Bayesian detector on top of Petri nets. It identified implausible co-durations and provided interpretable, structure-driven signals [10]. In the same "time as a first-class citizen" vein, Pereira and Varajão organized absolute and relative constraints, enabling validation of executions against policies [11]. Later, Senderovich et al. proposed a temporal network based on Allen's algebra. It unified control-flow and performance, avoiding the biases of analyzing them separately [12]. Subsequently, Zhao and Zhou extended modeling with TH-BPM. They added granularities and parameterized signatures to accept or flag deviations with traceability [20].

In parallel, data-driven approaches emerged. Sontheim et al. defined trace-level deviation profiles, which facilitated comparison and grouping of executions [14]. When latencies did not follow simple distributions, Mavroudopoulos and Gounaris proposed distance-based methods that were more sensitive to such cases [17]. For collective anomalies, Richter et al. developed TOAD, which ordered traces and detected micro-accumulations that are hard to see on a per-case basis [21]. For concurrency, Böhmer and Rinderle-Ma addressed multi-instance settings [13]. In evolving contexts, Richter and Seidl presented TESSERACT to detect temporal drifts in streaming, with alerts and threshold readjustment [16]. These techniques did not infer causality explicitly. Even so, they provided signals indicating where to experiment and intervene.

Several lines then connected detection and action. Saralaya et al. studied temporal impact in service-based systems and proposed SLA-oriented adaptations [15]. Chapela-Campa and Dumas modeled extrinsic delays in simulation, clarifying sources not attributable to resources [18]. To communicate patterns, Magallanes et al. offered multi-level visualization [19]. Ouarhim et al. proposed continuous improvement with right-real-time validation [22]. Taken together, these contributions broadened temporal interpretability. Causal explainability remained open. This work leverages those signals to bridge temporal diagnosis and causal decision-making.

2.2 Causality in Business Processes

In the causal line, Qafari and van der Aalst proposed an SEM-based scheme for root-cause analysis and developed a ProM plugin. The approach discovers the causal structure and estimates intervention ("*what-if*") effects on process indicators. The equational form and associated DAG provide high interpretability. The cost is strong assumptions: causal sufficiency, independence of noise terms, among others [23]. As an extension, they incorporated feature recommendation, including temporal aggregations, thereby reducing the causal space and simplifying the SEM. Readability improved while preserving intervention simulation. They identified "more understandable" diagnostics as future work, emphasizing practical interpretability over mere technical capacity to infer causes [24]. In a complementary approach, Greg Van Houdt et al. presented AITIA-PM. It formulates causal hypotheses using probabilistic temporal logic (*leads-to with windows*), then tests those hypotheses with significance tests. The output consists of probabilistic temporal rules, immediately readable by the analyst. However, it does not construct a full structural model of the process [26].

From operations-oriented discovery, Zhiwei Xing et al. integrated process mining with local graph learning. They used SMMB under MAG to handle latent variables, identified airport service factors associated with delay, and marked bidirectional edges where unobserved confounders were present. This helped prioritize actions. The explanation remained at the level of local adjacencies, with less detail on effect magnitudes and counterfactuals [25]. Finally, Zahra Dasht Bozorgi et al. combined causal inference with reinforcement learning. They estimated CATE with intervals and applied conformal prediction. The intervention decision was grounded in the estimated effects and achieved faster convergence. The learned policy, however, retained the inherent opacity of RL, so explainability remained tied to signals (CATE *and conformal sets*) rather than a detailed causal narrative [27].

3 Proposed Method

We propose a method for the causal analysis of temporal deviations at the level of process variants (*see* Fig. 2). Unlike approaches that treat all traces jointly (Section 2.1), we work per variant, and, when appropriate, per clusters with similar temporal profiles, to mitigate Simpson's paradox and reduce biases from global signatures such as those of Böhmer and Rinderle-Ma [13] and Sontheim et al. [14]. The pipeline consists of: ***(i)*** extracting variants from the log; ***(ii)*** constructing the temporal spectrum and clustering; and ***(iii)*** performing causal analysis per variant-cluster with interpretability and explainability metrics, so that the identification of causal relationships is grounded in the objectives of each execution pattern.

3.1 Event Log Derived from Planning

In this Stage (*see* Fig. 2) we propose deriving an event log from the project plan in order to establish a temporal baseline on which to capture process variants [28]. We represent the planned log L^{plan} as a multiset of traces σ (see).

Each trace $\sigma^j = \langle e_1^j, \ldots, e_{n_j}^j \rangle$ describes a case in the plan, and each event

$$e_k^j = \langle case^j,\ a_k,\ [t_{s,k}, t_{e,k}],\ \mathbf{x}_k \rangle \tag{2}$$

Contains the planned activity a_k, its time interval $[t_{s,k}, t_{e,k}]$, and an operational-variables vector $\mathbf{x}_k$. In a planning context, $\mathbf{x}$ might include, for example, the assigned resource or role, priority, expected duration, predecessor dependency, estimated cost, and load metrics. From L^{plan} we discover the model [29] and, above all, extract the set of process variants $V = \{v_i\}_{i=1}^{k}$ where each v_i is the canonical sequence of activities anticipated for a subset of cases. This variant-based representation allows us to reflect the plan's heterogeneity and anchors the next step in the workflow: constructing the spectrum and grouping similar variants (3.2), before applying causal analysis within each group (3.3).

Working with the log derived from the plan makes explicit the temporal commitments and dependency relationships encoded at design time, thereby facilitating the identification of temporal deviations and the interpretation of their impact in terms of schedule and planned resources [11].

3.2 Temporal Vectors and Variant-Level Clustering

In Stage 3.2, starting from the variants obtained in 3.1, we construct, for each process variant $V_j = \langle \sigma_i \rangle_{i=1}^{n}$, a temporal vector per trace. For a trace $\sigma \in \Sigma(V_j)$, we compute the intervals between consecutive activities $\Delta t_\ell^\sigma = t(a_{\ell+1}) - t(a_\ell)$ and normalize each interval using the modified z-score $z_m(x) = 0.6745\,\frac{x-\tilde{x}}{\text{MAD}(x)}$, where $\tilde{x}$ is the median of that interval over V_j and $\text{MAD}(x) = \text{median}(|x - \tilde{x}|)$. We use the modified z-score, rather than the standard one, because real-world log durations tend to be skewed and contain outliers; the median and MAD are robust to these effects, whereas the mean and standard deviation in the classic z-score can be distorted.

The temporal vector of σ is thus

$$z_j^\sigma = \left[z_m(\Delta t_1^\sigma), \ldots, z_m(\Delta t_{n-1}^\sigma) \right] \tag{3}$$

Using the set $\{z_j^\sigma\}_{\sigma \in \Sigma(V_j)}$, we perform variant-level clustering via a Gaussian Mixture Model (GMM), which groups traces with similar temporal patterns without assuming strict linearity or separability [30]. This step distinguishes typical behaviors from deviation patterns within the same variant.

We then select the cluster of interest as the one exhibiting appreciable internal variability and high average displacements $|\bar{z}|$ across multiple components, indicators of sustained deviation. We project its cases back onto the original log and apply the Performance Spectrum over $\{a_i\}_{i=1}^{n}$ of V_j: we compare the cluster's timings against its variant baseline to identify activities whose segments display shifted or elongated bands relative to the standard.

We flag for causal interpretation those activities that satisfy both ***(i)*** $|z_j^\sigma| > \tau$ in a significant fraction of cluster cases (with typical threshold $\tau \in [2, 2.5]$), and ***(ii)*** persistent displacement in the spectrum (*e.g., above the 90th percentile for the variant*).

3.3 Causal Interpretability and Explainability

In this stage, for each variant V_j and cluster $C_{j,\ell} \subseteq V_j$ (*from* 3.2 *via* z_j), we select focal activity a^* and estimate a consensus DAG guided by domain knowledge using the sub-log of $C_{j,\ell}$. With this DAG, we interpret temporal deviations of a^* and quantify interventions on operational log variables. Unlike SEM on the full process, our variant–cluster scope avoids blending variants, reduces aggregation bias, and enhances causal interpretability while remaining compatible with SEM in process mining [24].

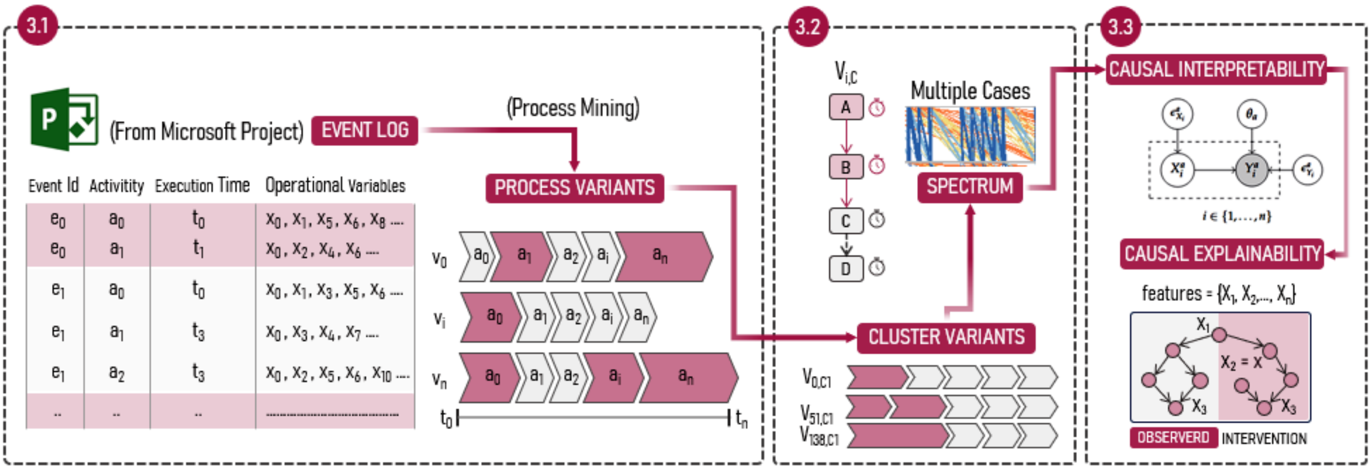

Figure 2: Scheme of the proposed methodology for analyzing underlying causal relationships in temporally clustered deviations from a process variant through variants, spectrum, and causal analysis.

3.3.1 Causal Interpretability. In this step, the causal structure of each cluster $C_{j,\ell}$ is estimated from its operational-variables matrix $\mathbf{D}_{j,\ell} \in \mathbb{R}^{n \times p}$, whose vertices are $V = \{X_i\}_{i=1}^{p}$. By means of a consensus among the algorithms $\mathcal{M}$: `LiNGAM, PC, GES` [7], the consensus graph E^* is defined as

$$E^* = \arg\max_{E \subseteq V \times V} \sum_{(i,k) \in E} f_{ik}$$

Where $f_{ik} = \frac{1}{|\mathcal{M}|} \sum_{m \in \mathcal{M}} \mathbf{1}\{ i \rightarrow k \in E^{(m)} \}$, is the vote frequency of each arc across the methods in $\mathcal{M}$, subject to the constraints $D_{\text{req}} \subseteq E$, $E \cap D_{\text{frb}} = \varnothing$, (*domain knowledge as required/forbidden directed arcs*), ensuring a robust estimate that aligns with the prior information.

Algorithm 1: Consensus Causal Graph Estimation

Input : $\mathbf{D}_{j,\ell} \in \mathbb{R}^{n \times p}$, $D_{\text{req}}, D_{\text{frb}} \subseteq V \times V$, $\mathcal{M} = \{$`LiNGAM`, `PC`, `GES`$\}$
Output: $\mathcal{G}^* = (V, E^*)$ with $V = \{X_1, \ldots, X_p\}$

for each $m \in \mathcal{M}$ **do**
  $E^{(m)} \leftarrow$ edges discovered by method m on $\mathbf{D}_{j,\ell}$
for each $(i,k) \in V \times V$ **do**
  $f_{ik} \leftarrow \frac{1}{|\mathcal{M}|} \sum_{m \in \mathcal{M}} \mathbf{1}\{ i \rightarrow k \in E^{(m)} \}$
$E^* \leftarrow \arg\max_{E \subseteq V \times V} \sum_{(i,k) \in E} f_{ik}$ subject to $D_{\text{req}} \subseteq E$, $E \cap D_{\text{frb}} = \varnothing$
while $\mathcal{G}^* = (V, E^*)$ *contains cycles* **do**
  $(i^*, k^*) \leftarrow \arg\min_{(i,k) \in \text{cycle} \setminus D_{\text{req}}} f_{ik}$
  $E^* \leftarrow E^* \setminus \{(i^*, k^*)\}$
return $\mathcal{G}^* = (V, E^*)$

note: An arc is included when its vote frequency $f_{ik} \geq 0.55$. Cycles are resolved by removing the non-required arc with the smallest f_{ik}. Hyperparameters: `PC` $\alpha = 0.05$; `GES`: `BIC` score with penalty $\lambda = 2$; `LiNGAM`: `DirectLiNGAM`.

From the perspective of causal interpretability, the obtained causal graph allows each arc $i \longrightarrow k$ to be understood as the direct influence of variable X_i on X_k, revealing the interaction dynamics among variables and facilitating comprehension of how potential interventions would modify the cluster's temporal behavior.

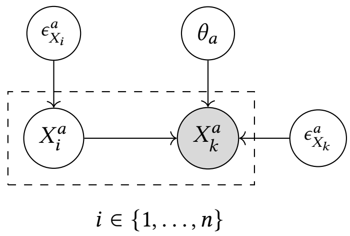

Figure 3: Causal model for activity a showing operational variable X_i^a influencing outcome X_k^a, with activity parameter θ and unobserved noise variables $\epsilon X i^a$, $\epsilon X_k{}^a$.

3.3.2 Causal Explainability. With $\mathcal{G}^*_{j,\ell}$ fixed for variant V_j and cluster $C_{j,\ell}$, we move from structural analysis to *what-if* scenarios, following the formulation of the do$(\cdot)$ – *operator* in Pearl's structural causal models [31, 32]. We select a treatable variable T in the context of the focal activity a^* and the outcome $Y \equiv Y_{a^*}$. The basic query is $\mathbb{E}\big[Y \mid \text{do}(T = t)\big]$,

$$\Delta(t, t_0) = \mathbb{E}\big[Y \mid \text{do}(T = t)\big] - \mathbb{E}\big[Y \mid \text{do}(T = t_0)\big].$$

For example, if T is the number of testers and $t_0 = 3$, comparing do$(T = 5)$ versus do$(T = 3)$ might yield

$$\Delta(5, 3) = \mathbb{E}\big[Y \mid \text{do}(T = 5)\big] - \mathbb{E}\big[Y \mid \text{do}(T = 3)\big] = -2 \text{ days},$$

indicating that adding two testers reduces the average approval time of a^* by two days. The adjustment set Z is chosen from $\mathcal{G}^*_{j,\ell}$ using the back-door criterion. Hence,

$$\mathbb{E}\big[Y \mid \text{do}(T = t)\big] = \sum_z \mathbb{E}\big[Y \mid T = t, Z = z\big] P(Z = z).$$

Figure 4 illustrates the mechanism. In the observed panel, C may influence both T and Y (edges γ, δ), and T influences Y (edge β). In the intervened panel do$(T = t)$, incoming edges to T are removed, isolating the direct effect β on Y. The analyst then compares $\mathbb{E}[Y \mid \mathrm{do}(T=t)]$ with the baseline t_0 and decides where and how much to intervene within $(V_j, C_{j,\ell})$ on the activity a^* that the performance spectrum marked as anomalous.

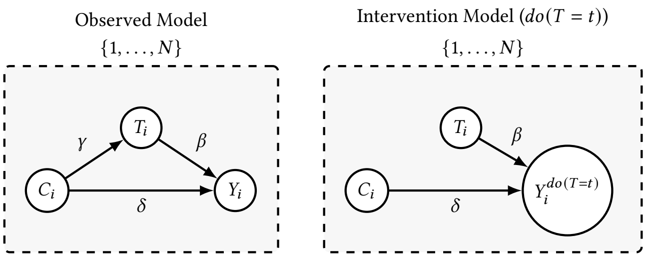

Figure 4: Causal model: in the *interventionist* scenario, do(T=t) removes the influence of C_i on T_i, thereby isolating the direct effect of T_i on Y_i.

4 Results

4.1 Dataset Description

An event log was derived from change plans and projects on products (*components, platforms, and services*) that included evolutionary developments and the resolution of incidents subject to planning. The log covered 1,261 traces (*6,015 events*), with eight standardized activities distributed across six variants.

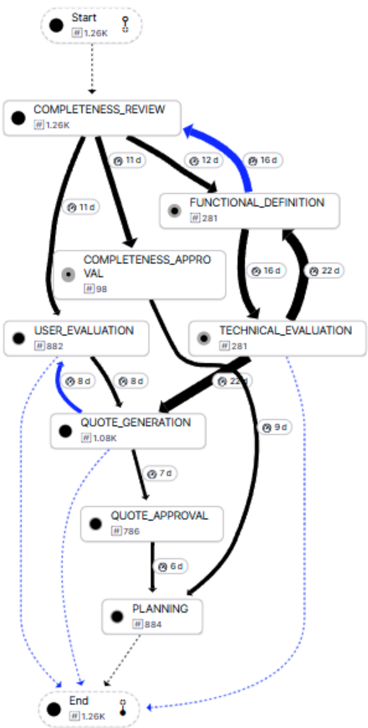

Figure 5: Executed process extracted from the event log and displayed as observed. It shows activities and their frequencies; generated with the Celonis Platform (https://www.celonis.com).

Each trace had a median of five events and a median cycle time of 46.6 days, measured from the first start timestamp to the last end timestamp. Five roles and sixteen users participated during the 2023–2024 period. Rework was observed in 22.2% of the traces, consistent with negotiation and adjustment iterations inherent to change management.

Table 1: Characteristics of the event log.

Metric	Value
Traces (cases)	1,261
Events	6,015
Activities (distinct)	8
Variants (distinct)	6
Dominant variant (ID, proportion)	3 (62.3%)
Events per trace (median)	5
Trace duration (median, days)	46.6
Traces with rework (%)	22.2
Roles (distinct)	5
Users (distinct)	16

4.2 Executed model and variants

The executed model was reconstructed by integrating the paths observed in the event log. From a process mining perspective, these paths, referred to as variants, represent complete sequences of activities that, when aggregated, allow visualization of the global structure of the process (*see* Fig. 5). Six variants were identified in total with the following frequencies: V_1 (98 traces), V_2 (97 traces), V_3 (786 traces), V_4 (85 traces), V_5 (99 traces) and V_6 (96 traces), totaling 1,261 recorded cases. According to this distribution, variant V_3 accounted for 62.3% of the traces analyzed, making it the most frequent route in the reconstructed model. To justify selecting variant V_3 as the main focus of analysis, both its numerical representativeness and its temporal profile were considered. When analyzing the intervals between consecutive activities within V_3 (n = 786), the temporal characterization summarized in Table 3 was obtained, organized by median (days).

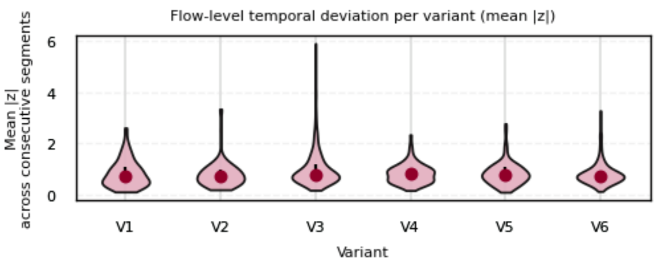

Figure 6: Flow-level temporal deviation per variant (median $|z|$).

Table 2: Process variants statistics, ordered by median.

Variant	n traces	Mean	Median	p75	p90	Max
4	85	0.829	**0.805**	1.016	1.260	2.335
3	786	**1.018**	0.789	**1.204**	**1.825**	**5.909**
5	96	0.821	0.788	1.056	1.198	2.771
6	99	0.819	0.724	0.896	1.247	3.275
2	97	0.796	0.720	0.974	1.189	3.342
1	98	0.830	0.715	1.090	1.596	2.607

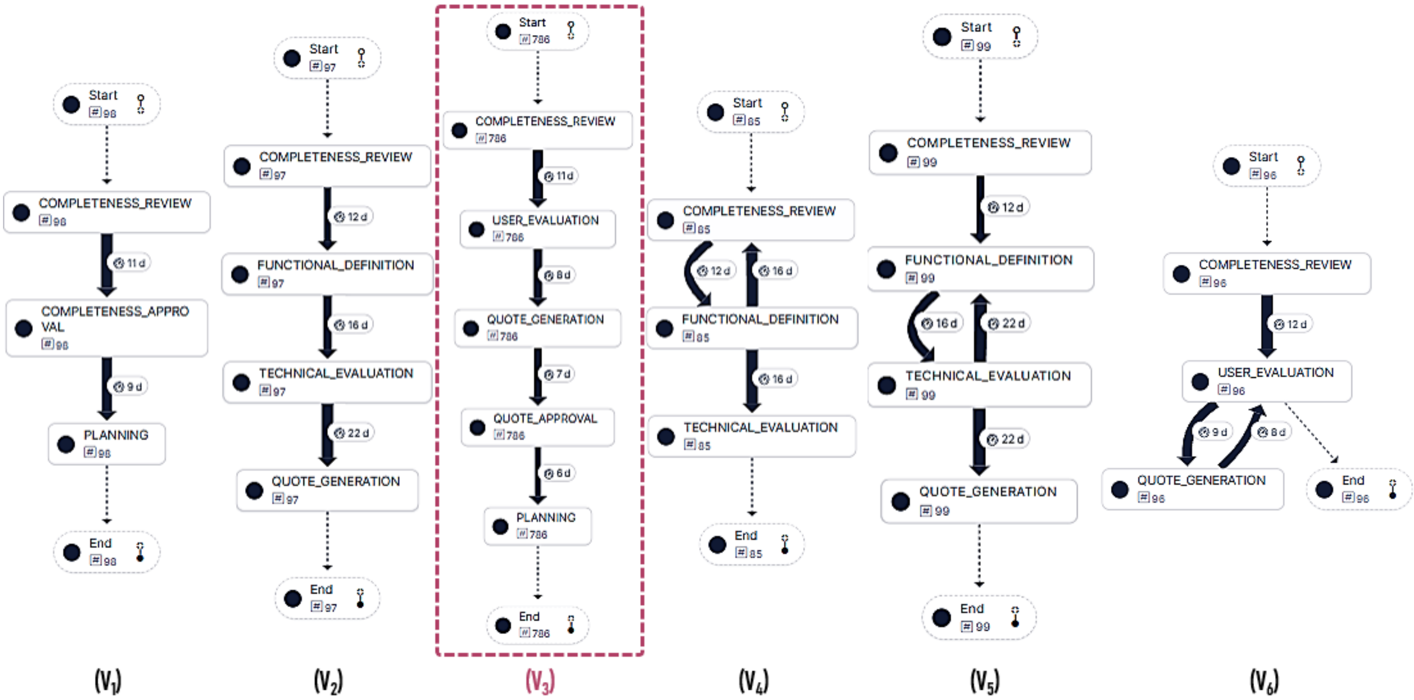

Figure 7: Six variants were extracted from the execution model (Celonis). Variant 3, highlighted, was selected for its predominance (62.3% of cases) and for exhibiting the temporal pattern of interest.

4.3 Process Variant

Variant 3 was selected based on its representativeness (786 of 1,261 traces, ≈ 62.3%) and its temporal profile, characterized by a decreasing gradient of delays between consecutive activities. The first transition (COMPL-REV ⟶ USER-EVAL) exhibited the greatest delay (median = 10.64 days; p90 = 16.39; SD = 4.42), while subsequent transitions had smaller medians (8.05; 7.36; and 5.62 days). Time normalization (original time, z-score, and MAD adjusted z-score) confirmed higher dispersion in this initial transition.

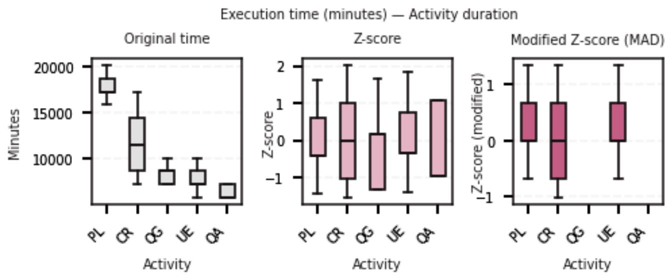

Figure 8: Execution-time distributions by activity: minutes (left), standard Z-score (center) and modified Z-score (MAD, right)

Table 3: Transition statistics (days), ordered by median.

Segment	Med.	Mean	p75	p90	Std
COMPL-REV ⟶ USER-EVAL	10.64	11.49	13.17	16.39	4.42
USER-EVAL ⟶ QUOTE-GEN	8.05	9.39	10.62	14.60	4.61
QUOTE-GEN ⟶ QUOTE-APP	7.36	8.10	8.72	11.03	2.83
QUOTE-APP ⟶ PLANNING	5.62	6.05	6.58	7.89	2.02

4.4 Cluster

Applying the GMM to variant 3 identified two subgroups: Cluster 0 (183 traces, 23.3%) and Cluster 1 (603 traces, 76.7%). Variability analysis revealed higher heterogeneity in Cluster 0 (total variance = 4.44; intra-case = 2.60; inter-case = 1.84). Its dispersion in PCA space confirmed atypical temporal patterns in the Δt vectors. Given this variability, Cluster 0 was prioritized for causal analysis, emphasizing extreme cases and their conditions. A cluster comparison was proposed to distinguish consistent patterns from episodic noise.

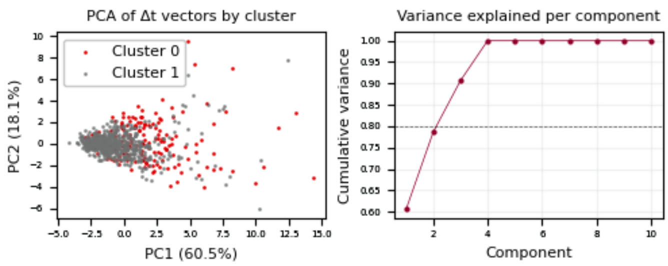

Figure 9: PCA of Δt vectors by cluster (left; PC_1 60.5%, PC_2 18.1%) and cumulative variance explained (right) (selected $k = 2$ by minimum BIC.)

4.5 Performance Spectrum

The Performance Spectrum for variant 3 identified the USER-EVAL activity as the primary delay point. Using quartile segmentation (Q3=5.2 days), critical cases (Q4) comprised 197 of 786 traces (25.1%) with delays from 5.2–29.5 days (mean=10.1 days), versus normal cases (mean=2.1 days). Critical delays appeared recurrently without clear seasonal patterns (max. 4 cases/month), indicating a persistent structural bottleneck. Delays at `USER-EVAL` directly affected the next transition (`USER-EVAL`⟶`QUOTE-GEN`), second slowest (median=8.05 days), amplifying downstream durations. Thus, `USER-EVAL` was prioritized due to high frequency, significant delays (8 extra days vs. normal), and temporal persistence. Extreme cases (e.g., case 820=29.5 days; case 499=27.6 days) provided specific instances to examine underlying causes (capacity, reviewer availability, or missing inputs) for targeted interventions.

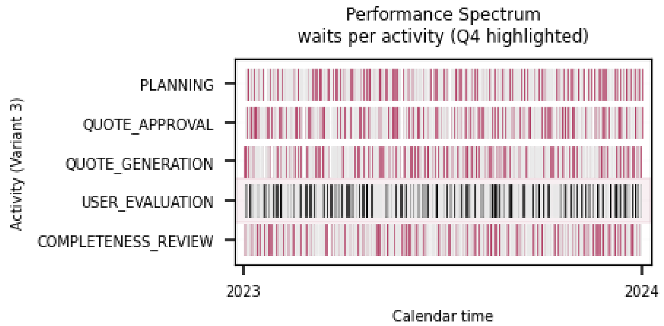

Figure 10: Performance Spectrum — Variant 3. Vertical lines = activity start times per case; cases with critical delays (Q4, >5.2 d) highlighted in purple (n = 786 traces).

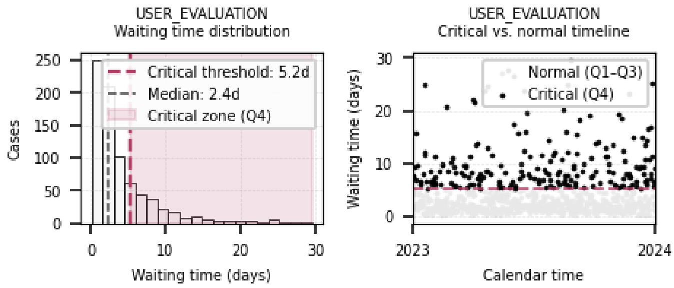

Figure 11: Waiting-time distribution at `USER-EVAL` (left) and timeline of critical (Q4, >5.2 d) vs normal cases (right).

The Performance Spectrum operated as a bridge between descriptive diagnosis and causal analysis: it enabled defining a prioritization criterion that weights (*i*) recurrence of the deviation, (*ii*) magnitude of the wait, and (*iii*) downstream propagation in the flow. Applied to activities, `USER-EVAL` concentrated the greatest impact, consistent with a plausible accumulation mechanism induced by complexity and volume of requirements that activates testing hours and coordination. This yields operational, testable hypotheses: set thresholds of *numrequirements* per service; monitor the ratio (*developmenthours* + *testinghours*) / *developers* against the *baseline*; and limit *nummeetings* per unit of effort. Framed this way, the spectrum not only pinpoints the bottleneck but also informs the interventions that are subsequently evaluated as $do(\cdot)$ in the causal explainability stage.

4.6 Causal Interpretability

Applying the causal interpretability phase to the critical activity `USER-EVAL` produced a causal graph that explains the operational sources of temporal variability (*see* Fig. 12), constructed under a transparent consensus criterion described below. Variables derived directly from the enriched process log reflect attributes specific to the execution of this activity (*see* Sec. 3.1). The graph revealed that requirement complexity (*complexitylevel*) causally determines the number of requirements (*numrequirements*) and the number of testers (*numtesters*), directly influencing the duration and delays associated with the evaluation.

The selection of activity `USER-EVAL` responded to its critical role within the persistent bottleneck identified in Variant 3, whose heterogeneity manifested clearly in the robust shifts observed ($|z| > \tau$, where τ denotes the robust cutoff with $\tau \in [2, 2.5]$) and persistence above the p_{90} of the variant *baseline*. This focused analysis preserves the process' inherent variability and enables a localized interpretation of the graph, thus facilitating specific and detailed explanations of the operational sources of temporal variability.

From this more granular perspective, it is also observed that the number of requirements causally affects the hours allocated to development (*developmenthours*) and to testing (*testinghours*), the latter directly influencing the number of meetings required (*nummeetings*). This causal structure clarifies how interactions among operational attributes generate the observed temporal variability, clearly highlighting the underlying causes of the critical delays in `USER-EVAL`.

The obtained graph was estimated by *consensus* among `LiNGAM`, `PC`, and `GES`, incorporating domain knowledge via *required/forbidden* edges ($D_{\text{req}}, D_{\text{frb}}$). Each edge was retained when its *vote frequency* exceeded the threshold ($f_{ik} \geq 0.55$), resolving cycles by removing the non-required edge with the lower vote (Algorithm 1); the hyperparameters used are also reported (`PC`: ($\alpha = 0.05$); `GES`: `BIC` ($\lambda = 2$); `LiNGAM: DirectLiNGAM`). This procedure provides auditability of the decisions made, clarifying why each connection appears in the figure and how its causal orientation was determined.

In structural-model terms, each edge ($i \to k$) in the figure is interpreted as the *direct influence* of (X_i) on (X_k) in the focal activity, with specific parameters and unobserved noise terms (*see the schematic in* Fig. 3). This explicit correspondence between topology and mechanisms allows reading the graph beyond mere correlation, coherently identifying which variables will be candidates for interventions in the subsequent explainability phase.

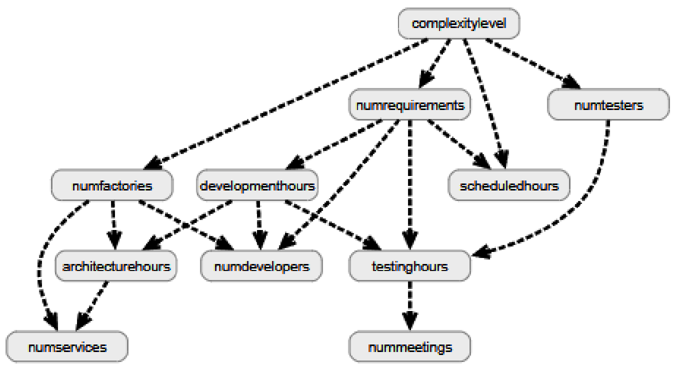

Figure 12: Causal graph obtained for the critical activity `USER-EVAL`, showing how specific operational attributes explain the observed temporal variability ($f_{ik} \geq 0.55$).

4.7 Causal Explainability

Table 4: Causal intervention analysis results.

Variable	Fact.	Int.	Δ	Δ%	d	Ben.
Arch. hours	27.21	29.06	1.85	6.8	0.40	No
Dev. hours	97.59	120.00	22.41	23.0	3.38	No
Developers	10.80	11.19	0.39	3.6	0.23	Yes
Meetings	10.94	12.25	1.31	12.0	1.26	No
Services	3.43	3.51	0.08	2.4	0.11	Yes
Test hours	44.59	51.17	6.57	14.7	1.50	No

note: d = Cohen's *d*; Ben. = beneficial for the objective (reduced duration/variability).

In this stage, an operational scenario defined by management was evaluated, aiming to slightly standardize the complexity level (*complexitylevel*) above the mean through functional agreements. The causal analysis showed a shift of the system toward higher demand, accompanied by growth in capacity (services, developers) and a more pronounced increase in effort and coordination (*developmenthours*, *nummeetings*), as evidenced in Figure 13.

The intervention $do(\text{complexitylevel} = \mu + \Delta)$ increased *developmenthours* ($+ \approx 19\%$), *nummeetings* ($+ \approx 17\%$), and *testinghours* ($+ \approx 21\%$), while the increases in developers and services were modest. The growth in *numrequirements* ($+ \approx 28\%$) acted as the main mediator of the effect, shifting the balance toward less favorable temporal performance in USER-EVAL, since aggregate effort rose faster than effective capacity (*see* Table 4).

The causal explanation is consistent with the graph structure: *complexitylevel* ⟶ *numrequirements*, *numtesters*; then *numrequirements* ⟶ *developmenthours*, *testinghours*; and finally *testinghours* ⟶ *nummeetings*. These pathways extended the effective duration of USER-EVAL and amplified process queues, confirming propagation of the effect through effort-related nodes. Estimates were obtained using the adjustment set derived from the DAG (back-door criterion), which permits a valid causal interpretation under the model assumptions.

In the evidence shown in Figure 13, the density panels and difference plots indicate that the series associated with capacity register useful increases, although insufficient to absorb the rise in *numrequirements*. The evidence in Table 4 reinforces this asymmetry: the larger increases in *developmenthours*, *testinghours*, and *nummeetings* substantially exceed the increases in developers and services. Operationally, a higher level of "green" does not necessarily imply temporal improvement when capacity grows below the demand to be processed.

As an additional verification, results held when varying the adjustment set within the DAG limits, reweighting by service size, and excluding effort outliers. The mediation decomposition pointed to a total effect dominated by the pathway *complexitylevel* ⟶ *numrequirements* ⟶ (*developmenthours*, *testinghours*), with a bounded residual direct effect. Concurrently, the absorption ratios (*developmenthours* + *testinghours*)/*developers* were consistently above the *baseline* under the intervention, which is consistent with the deterioration observed in temporal performance.

In summary, increasing *complexitylevel* above the mean without prior constraints generated an imbalance between capacity and demand. *Numrequirements* emerged as the dominant mediator, driving *developmenthours*/*testinghours* and increasing *nummeetings*. Consequently, decisions should be conditioned on mediator-containment metrics (e.g., thresholds of *numrequirements* per service) and on capacity absorption ratios (e.g., (*developmenthours* + *testinghours*)/*developers* relative to the *baseline*). The results summarized in Figure 13 and Table 4 show that these ratios deteriorated under the evaluated intervention, so the policy was not effective in reducing temporal variability in USER-EVAL.

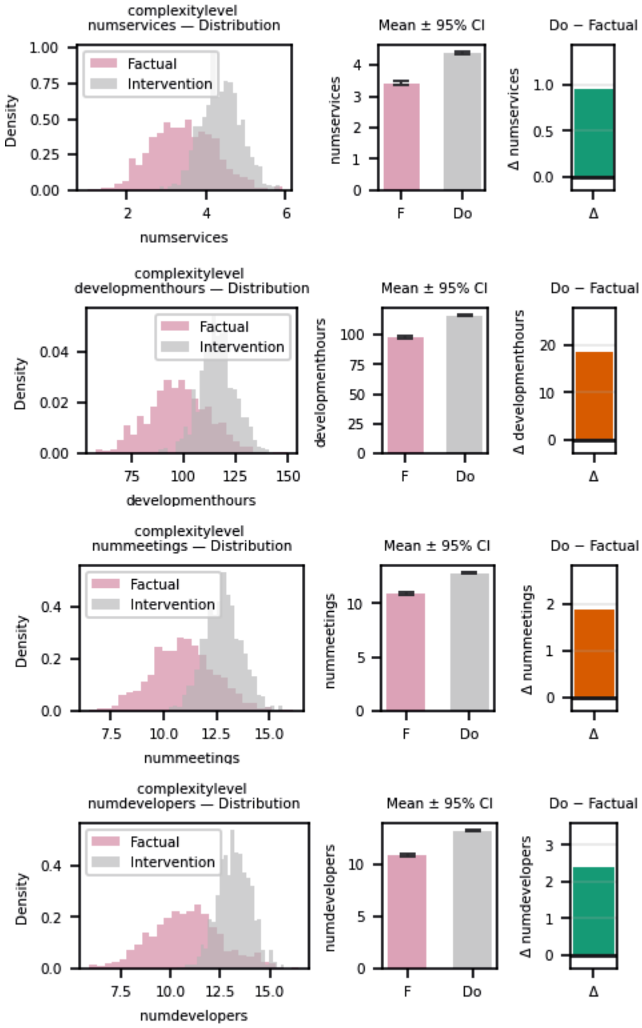

Figure 13: Causal analysis of the $do(\text{complexitylevel} = \mu + \Delta)$ intervention on key operational variables. Distributions (*factual vs. intervention*) and resulting differences (Δ) are compared to assess the impact on temporal variability.

5 Discussion

The analysis confirmed that USER-EVAL concentrates significant temporal deviations ($Q_4 > 5.2$ days, mean 10.1 days), causing persistent downstream bottlenecks. Additionally, other transitions, such as COMPL-REV ⟶ USER-EVAL (median 10.6 days), showed a substantial accumulation of minor delays, globally affecting process performance (RQ2).

These findings align with previous studies highlighting the importance of managing critical stages with concentrated delays. For example, Böhmer and Rinderle-Ma [13] use temporal signatures to detect early anomalies, while Sontheim et al. [14] identify deviations through detailed analyses of specific intervals between activities. Although these methods face challenges in extensive processes, they underscore the relevance of effectively identifying and managing critical points (RQ2).

Validating our method using a log derived from planning artifacts (1,261 traces, 6,015 events, 8 activities, 6 variants) and focusing the analysis on specific activities strengthens the reliability of these results. However, the use of a single dataset limits generalizability; future studies could replicate this analysis in different organizational contexts or extend the observation period to assess result stability (RQ1).

Future research could also examine organizational factors such as resource allocation or automation technologies to evaluate how these interventions influence effective reductions in temporal deviations. This would improve understanding of the causal link between managerial interventions and operational efficiency (RQ3).

In summary, the precise identification of critical activities and their causal relationships provides a practical foundation for targeted interventions. Addressing these points directly offers genuine potential for enhancing operational performance and optimizing overall process management (RQ3).

6 Conclusions

This work integrated process mining and causal learning techniques to identify and explain temporal deviations in business process flows. The proposed methodology combines causal discovery algorithms with event-level analysis, enabling not only process interpretability but also concrete causal explanations of observed deviations. Specific activities responsible for significant delays were identified, and causal interventions quantified the impact of variables such as scheduled hours, complexity level, and number of requirements on overall process duration.

Applying causal models proved effective for guiding precise operational decisions, though extending the generalizability of these results will require analyzing broader datasets and diverse organizational contexts. Future research could further explore how resource allocation and automation technologies help reduce temporal deviations. Ultimately, integrating causality into process mining provides practical tools for optimizing operational management, reducing execution times, and improving process planning efficiency.

References

[1] Fischer, D. A., Marcus, L., & Röglinger, M. (2024). A portfolio management method for process mining-enabled business process improvement projects. Business & Information Systems Engineering.

[2] Di Francescomarino, C., Donadello, I., Ghidini, C., Maggi, F. M., & Puura, J. (2025). Business process deviance mining with sequential and declarative patterns. Business & Information Systems Engineering.

[3] Fournier, F., Limonad, L., Skarbovsky, I., & David, Y. (2025). The WHY in business processes: Discovery of causal execution dependencies. KI – Künstliche Intelligenz.

[4] Negri, F. (2023). Correlation is not causation, yet… Matching and weighting for better counterfactuals. En A. Damonte & F. Negri (Eds.), Causality in Policy Studies: A pluralist toolbox (pp. 71–98). Springer.

[5] Kone, M.H. (2025). Correlation Versus Causation. In: Lovric, M. (eds) International Encyclopedia of Statistical Science. Springer, Berlin, Heidelberg.

[6] Wang, L., Huang, S., Wang, S., Liao, J., Li, T., & Liu, L. (2024). A survey of causal discovery based on functional causal model. Engineering Applications of Artificial Intelligence, 133, 108258.

[7] Zanga, A., Ozkirimli, E., & Stella, F. (2022). A survey on causal discovery: Theory and practice. International Journal of Approximate Reasoning, 151, 101–129.

[8] Denisov, V., Fahland, D., & van der Aalst, W. M. P. (2018). Unbiased, fine-grained description of processes performance from event data. En Business Process Management – BPM 2018. Springer.

[9] Denisov, V., Fahland, D., & van der Aalst, W. M. P. (2020). Repairing event logs with missing events to support performance analysis of systems with shared resources. En Application and Theory of Petri Nets and Concurrency – Petri Nets 2020. Springer.

[10] Rogge-Solti, A., Kasneci, G. (2014). Temporal Anomaly Detection in Business Processes. In: Sadiq, S., Soffer, P., Völzer, H. (eds) Business Process Management. BPM 2014. Lecture Notes in Computer Science, vol 8659. Springer, Cham.

[11] J. L. Pereira and J. Varajão, "The Temporal Dimension of Business Processes - Dealing with Time Constraints," Procedia Computer Science, vol. 121, pp. 1034-1038, 2017.

[12] Senderovich, A., Weidlich, M., Gal, A. Temporal Network Representation of Event Logs for Improved Performance Modelling in Business Processes. In: Carmona, J., Engels, G., Kumar, A. (eds) Business Process Management. BPM 2017. Lecture Notes in Computer Science, vol 10445. Springer, Cham.

[13] Böhmer, K., & Rinderle-Ma, S. Multi instance anomaly detection in business process executions. In Business Process Management: 15th International Conference, BPM 2017, Barcelona, Spain, September 10–15, 2017, Springer.

[14] Sontheim, J., Richter, F., Seidl, T. : Temporal deviations on event sequences. In: LWDA (2019).

[15] Sridevi Saralaya, Rio G. D'Souza, and Vishwas Saralaya. 2019. Temporal impact analysis and adaptation for service-based systems. Int. J. Inf. Commun. Techol. 14, 4 (2019), 403–426.

[16] Florian Richter, Thomas Seidl, Looking into the TESSERACT: Time-drifts in event streams using series of evolving rolling averages of completion times, Information Systems, Volume 84, 2019, Pages 265-282, ISSN 0306-4379.

[17] Mavroudopoulos, I., & Gounaris, A. (2020). Detecting temporal anomalies in business processes using distance-based methods. In Discovery Science: 23rd International Conference, DS 2020, Thessaloniki, Greece, October 19–21, 2020, Proceedings 23 (pp. 615-629). Springer International Publishing.

[18] David Chapela-Campa and Marlon Dumas. 2024. Enhancing business process simulation models with extraneous activity delays. Inf. Syst. 122, C (May 2024).

[19] Magallanes, J., Stone, T., Morris, P. D., Mason, S., Wood, S., & Villa-Uriol, M.-C. (2022). Sequen-C: A multilevel overview of temporal event sequences. IEEE Transactions on Visualization and Computer Graphics, 28(1), 901–911.

[20] D. Zhao and Z. Zhou, "Anomaly and Interestingness Detection in Timed Hierarchical Business Processes," in IEEE Transactions on Engineering Management, vol. 71, pp. 12619-12634, 2024.

[21] Richter, F., Lu, Y., Zellner, L., Sontheim, J., & Seidl, T. (2020, October). TOAD: trace ordering for anomaly detection. In 2020 2nd International Conference on Process Mining (ICPM) (pp. 169-176). IEEE.

[22] Asma Ouarhim, Karim Baina and Brahim Elbhiri, "Towards a Continuous Temporal Improvement Approach for Real-Time Business Processes" International Journal of Advanced Computer Science and Applications(IJACSA), 15(1), 2024.

[23] Qafari, M.S., van der Aalst, W. Root Cause Analysis in Process Mining Using Structural Equation Models. In: Del Río Ortega, A., Leopold, H., Santoro, F.M. (eds) Business Process Management Workshops. BPM 2020. Lecture Notes in Business Information Processing, vol 397. (2020) Springer, Cham.

[24] Qafari, M.S., van der Aalst, W.M.P. Feature recommendation for structural equation model discovery in process mining. Prog Artif Intell (2022).

[25] Luo, Qian, Zhang, Lin, Xing, Zhiwei, Xia, Huan, Chen, Zhao-Xin, Causal Discovery of Flight Service Process Based on Event Sequence, Journal of Advanced Transportation, 2021, 2869521, 17 pages, 2021.

[26] Van Houdt, G., Depaire, B., Martin, N. (2022). Root Cause Analysis in Process Mining with Probabilistic Temporal Logic. In: Munoz-Gama, J., Lu, X. (eds) Process Mining Workshops. ICPM 2021. Lecture Notes in Business Information Processing, vol 433. Springer, Cham.

[27] Bozorgi, Z.D., Dumas, M., Rosa, M.L., Polyvyanyy, A., Shoush, M., Teinemaa, I. (2023). Learning When to Treat Business Processes: Prescriptive Process Monitoring with Causal Inference and Reinforcement Learning. Advanced Information Systems Engineering. CAiSE 2023. Lecture Notes in Computer Science, vol 13901. Springer, Cham.

[28] Van Der Aalst, W., (2016). Data science in action (pp. 3-23). Springer Berlin Heidelberg.

[29] Ahmed, R., Faizan, M., Burney, A.I.: Process Mining in Data Science: A Literature Review. In: 2019 International Conference on Advances in the Emerging Computing Technologies (AECT), pp. 1–6. IEEE (2019).

[30] Sarang, P., Gaussian Mixture Model. In: Thinking Data Science. The Springer Series in Applied Machine Learning. (2023) Springer, Cham

[31] Pearl, J. (2009). Causality: Models, reasoning, and inference (2nd ed.). Cambridge University Press.

[32] Nogueira, A. R., Pugnana, A., Ruggieri, S., Pedreschi, D., & Gama, J. (2022). Methods and tools for causal discovery and causal inference. Wiley interdisciplinary reviews: data mining and knowledge discovery, 12(2), e1449.

Chapter 5
General Discussion

Esta tesis establece un avance metodológico en Minería de Procesos orientado a la gestión de desviaciones temporales. Al integrar la jerarquía causal de Pearl, la investigación supera las limitaciones de las técnicas puramente observacionales (L_1) y demuestra la viabilidad de niveles superiores de razonamiento para explicar y mitigar ineficiencias operacionales.

1. **Fundamentos estructurales ($L_1 \to L_2$):** El primer avance consistió en definir una representación estructural del proceso. Los trabajos [4.1] y [4.4] revelaron la topología causal (`DAGs`), base necesaria para cualquier intervención. El prototipo `CaProM` [4.4] demuestra su utilidad: el monitoreo ya no se limita a detectar desviaciones, sino que identifica su origen causal, lo que permite avanzar hacia modelos capaces de intervenir.
2. **Capacidad de intervención (L_2):** Sobre esta base, los artículos [4.3], [4.5] y [4.6] transformaron el modelo en una herramienta prescriptiva mediante el operador $do(x)$. La tesis demuestra que es posible estimar efectos antes de su ejecución. El análisis de efectos heterogéneos [4.6] permite identificar en qué contextos una intervención es más efectiva, optimizando así los recursos sobre una base causal.
3. **Razonamiento contrafactual (L_3):** Los trabajos [4.2] y [4.7] incorporaron la capacidad de responder a la pregunta: *"qué habría ocurrido si"*. Este tipo de razonamiento es clave en entornos industriales que requieren trazabilidad y explicaciones claras. Las explicaciones contrafactuales permiten auditar decisiones, asignar responsabilidades y generar recomendaciones transparentes. La propuesta no solo mejora la eficiencia operacional (L_2), sino que también habilita el aprendizaje retrospectivo del sistema (L_3).

Cada pregunta de investigación contribuye a un nivel de la jerarquía. RQ_1 valida las condiciones estructurales requeridas para avanzar más allá del nivel observacional (L_1). RQ_2 fortalece la interpretabilidad y consolida el nivel de intervención (L_2). RQ_3 aborda la explicabilidad de las estrategias de mejora mediante razonamiento contrafactual (L_3), permitiendo simular y auditar intervenciones antes de su aplicación.

5.1 RQ.1

To what extent do temporal deviations in process variants affect the identification and quantification of causal relationships among activities, resources, and roles in business processes?.

5.1.1 Temporal Disaggregation as a Requirement for Causal Validity

The results indicated that the validity of the inference depended on disaggregating by variants and temporal clusters, because aggregation mixes populations with different mechanisms and can induce Simpson's paradox. This pattern was consistent at the variant or cluster level and was not extrapolated beyond the group [4.1, sec.B], [4.4, sec.5.3], [4.5, sec.4.3]. Under assumptions of stability/SUTVA, ignorability after disaggregation, and positivity, and for the case analyzed, the triangulation of LiNGAM, PC, and GES identified a DAG more consistent than the one obtained when groups were not separated [4.1, sec.B], [4.4, sec.5.4], [4.7, sec.3.3]. The influence of latent confounders was not ruled out [4.1, sec.B], [4.7, sec.3.3]; consequently, explicit separation could partially reduce this uncertainty.

The methodology required explicit separation by variant or temporal cluster to mitigate aggregation bias ([4.1, sec.B], [4.5, sec.4]) and to isolate homogeneous subsets ([4.5, sec.4.3], [4.7, sec.3.2]). TD were identified using the Modified Z-Score (median, MAD), which is robust to heavy tails and outliers ([4.5, sec.4.2]). Segmentation was performed with GMM or spectral clustering in a multiview latent space (*flow, roles, and time*) ([4.7, sec.3.2], [4.6, sec.4.3]). Additionally, the *Performance Spectrum* identified anomalous focal activities to guide intervention ([4.7, sec.4.5], [4.7, sec.3.2]).

Given that time can act as a confounder or effect modifier (Chapter 2 secc(2.2.3.1)), organizational mining was incorporated as a link between time and structure. This approach characterized role and resource interactions (*handover, collaboration*), defined covariates, and specified adjustment sets prior to causal discovery [4.3, sec.B, sec.C], [4.7, sec.3.3.2]. First, the interaction topology was identified to isolate heterogeneity arising from work patterns (*delegation, reassignment*); subsequently, causal discovery was applied to operational variables [4.3, sec.B], [4.3, sec.C], [4.3, sec.A]. The reconstructed network made it possible to derive role and resource variables and integrate them into the structural consensus and the adjustment sets, which improved the interpretation of effects by subgroup [4.3, sec.B], [4.3, sec.C], [4.7, sec.3.3, sec.3.3.2].

Patterns of delegation and reassignment associated with changes in cycle time were observed, consistent with the influence of organizational structure on temporal deviations [4.3, sec.I-A]. For quantification, under identifiability assumptions, the analysis was conducted within the $(do(\cdot))$ framework [4.2]; group-specific CATE were estimated [4.6], [4.7, sec.3.3.2], *cross-fitting* with DML and ORF was employed, and confidence intervals were reported. The effect size was interpreted based on the confidence intervals and the sample size; with wide intervals or small samples, conclusions remained conditional [4.6, sec.4.4], Chapter 2 secc(2.2). In summary, disaggregation mitigated mixing bias and stabilized the orientation of causal arcs within specific subgroups, partially addressing RQ.1 in its structural and quantitative dimensions [4.2], [4.6, sec.4.4].

5.1.2 Related Work (Causal Integration)

In the literature on temporal deviations, the primary focus has been locating anomalies in L_1 (*associative*, *see* Table 2.1) through drift detection (TESSERACT) or outlier-cluster identification (TOAD, distances) [57], [62], [58]. In temporal representation (TNR, TH-BPM), control-flow is validated and modeled [53], [61], [65]. The compendium's findings aligned with that line and extended it by employing TD localization as a stratification criterion for causal discovery on operational variables (*roles, resources, complexity*) [4.7, sec.3.2], [4.7, sec.3.3], [4.3, sec.B, sec.C].

In process-applied causality, comparing with the literature contextualized the results and highlighted advantages of the compendium's approach for causal **identification** and **quantification**.

(***i***) First, compared with SEM on aggregated tables, the literature using *situation feature tables* models linearrelationships [67], [68], which can inherit biases in the presence of TD. In turn, after disaggregating by variant or cluster, PC (*conditional independences*), GES (*score-based*), and LiNGAM (*non-Gaussianity*) were applied [4.5, sec.4.3], [4.7, sec.3.3], [4.1, sec.B]; lower exposure to temporal mixing was observed, conditioned on the level of disaggregation and the assumptions of each algorithm.

(***ii***) Subsequently, regarding temporal precedence (*Granger*) versus interventional interpretation ($\mathrm{do}(\cdot)$, CATE), Granger evaluates whether the past of (X) improves the predictability of (Y) (*precedence*) but does not identify ($P(Y \mid \mathrm{do}(X))$) or control for confounding [72]. By contrast, in stratified DAGs, adjustment sets were identified and effects were estimated under ($\mathrm{do}(\cdot)$) with group-specific CATE using DML/ORF (*cross-fitting*) [4.7, sec.3.3.2], [4.6, sec.4.4], [4.2]; ignorability and positivity conditions were verified, and the effective sample size per group was controlled, enabling an interventional interpretation of the results.

(***iii***) In addition, regarding a single estimator versus cross-algorithm consensus, some approaches rely on a single causal-discovery method. By contrast, LiNGAM/PC/GES were triangulated, a consensus graph was constructed, and structural stability was assessed via *bootstrapping* [4.7, sec.3.3, subsec.3.3.1], [4.5, sec.4.3], [4.4, sec.5.4]; greater structural stability was demonstrated than with individual estimators, although instabilities persisted in arcs sensitive to sampling.

(***iv***) Likewise, regarding control-flow and temporal constraints versus operational levers in the DAG, TNR and TH-BPM represent the flow and validate temporal constraints [53], [61], [65]. Beyond that representation, role, resource, and complexity variables were integrated into the DAG as group-specific intervention levers [4.7, sec.3.2], [4.3, sec.B, sec.C], [4.4, sec.5.4]; thus, the causal structure was linked to operational decisions, according to the graph specification.

(***v***) Moreover, regarding isolated diagnostics versus an end-to-end route (*detection, explanation, and recommendation*), driftdetectors localize anomalies [57], [62]. Rather than stopping there, *Performance Spectra*, DAG, and CATE were chained [4.7, sec.4.5], [4.7, sec.3.3.2], [4.6, sec.4.4]; this made it possible to move from "*where and when*" to "*why*" and "*how much under intervention*."

(***vi***) As for assumptions, comparing assumptions implicit at the aggregate level versus assumptions made explicit by subgroups, in the compared literature assumptions remain implicit at the aggregate level. In contrast, identifiability assumptions (SUTVA, *ignorability, positivity*) and non-extrapolation were delineated explicitly by group Chapter 2 secc(2.2), [4.7, sec.3.3]; this specified the domain of validity of the effects.

(***vii***) In organizational mining, contrasting aggregate flow/SEM with a role-oriented approach, aggregate-flow or SEM approaches describe global structures [53], [61], [67], [68]. In contrast, *handover*networks were reconstructed and role-based metrics were derived to inform covariates and DAG constraints [4.3, sec.B, sec.C], [4.7, sec.3.2]; this linked organization and temporality.

(***viii***) Finally, regarding aggregate averages versus heterogeneity of effects (CATE), aggregate analyses conceal sign reversals. Accordingly, CATE were reported by group (coexistence of positive and negative effects) [4.7, sec.3.3.2], [4.6, sec.4.4]; this supported the prioritization of group-specific interventions, documented with confidence intervals and sensitivity analysis.

In terms of robustness, LiNGAM/PC/GES consensus (frequency voting with required-edges) and *bootstrapping* were applied to estimate edge stability; the results indicated greater structural coherence when stratifying [4.7, subsec.3.3.1], [4.5, sec.4.3], [4.4, sec.5.4]. During estimation, *cross-fitting* was used, acknowledging multiplicity and sample size by group; the presence of unobserved confounders, missing data, or measurement biases was not ruled out, which conditioned generalization on these factors Chapter 2 secc(2.2), [4.1, sec.B], [69].

Regarding scope, these conclusions were limited to the defined groups (variant or temporal cluster) and to the case study; extrapolation to other processes requires additional replication and sensitivity analysis to latent confounders [4.1, sec.B], [4.6, sec.4.4]. Taken together, the evidence suggests that temporal disaggregation could constitute an operational requirement to sustain causal validity in comparable contexts.

5.2 RQ.2

To what extent does causal interpretability contribute to the understanding of temporal deviations by providing quantitative representations and structured explanations of process flows?

5.2.1 Interpretable Directed Causal Topology

The results indicate that causal interpretability allows advancing from correlational diagnostics toward a directional topology (DAG) of operational variables [4.1], [4.2]. Under this perspective, a structured basis is provided for managing temporal deviations, revealing the causal hierarchy that conditions temporal effects and guiding managerial decisions [4.7, sec.3.3], [4.4, sec.5.4].

The causal topology operates as a central instrument. It establishes explicit directions: $(X \to Y)$ does not imply $(Y \to X)$, thereby overcoming the symmetry inherent in correlation [4.1]. As a case study, after disaggregation by variant–cluster, a consensus DAG was estimated. This DAG prioritized USER-EVAL as a critical delay point and allowed tracing propagation paths of deviations [4.7, sec.4.6]. This representation connects diagnostics with scenario evaluation ($\mathrm{do}(\cdot)$) [4.7, sec.4.5]. Furthermore, it enables prioritizing interventions on operational levers, roles (R_i), quantities (Q_i), and times (T_i), with explicit traceability of assumptions and identification [4.7, sec.3.3.2], [4.2], [4.3, sec.B, sec.C].

5.2.2 Related Work (Comparison of Causal Topologies)

(***i***) **From aggregated SEM/PAG to DAG per variant**. In contrast to aggregated-data SEM, which yields PAG (*Partial Ancestral Graphs*) and requires subsequent domain-driven orientation rules [67], [68], the compendium adopts (i) disaggregation by variant–cluster and (ii) a consensus-based, multi-method DAG. This consensus leverages arc frequency (arc $(i \to k)$) is retained if it exceeds a threshold among LiNGAM/PC/GES), complemented by uncertainty estimation through bootstrap entropy ($H(p_{i \to k})$). In [4.7, sec.3.3, sec.3.3.1], consensus with domain restrictions was applied for traceability; in [4.4, sec.5.4], unstable arcs were additionally filtered via entropy. This procedure avoids mixing mechanisms and provides an oriented, auditable topology to analyze temporal deviations. In contrast, SEM/PAG methods do not report multi-method consensus nor explicit arc uncertainty, limiting directional interpretation and interventions [67], [68].

(***ii***) **PTL rules versus global network of interdependencies**. PTL (*Probabilistic Temporal Logic*) enables formulating probabilistic-temporal hypotheses over event sequences ("*always*," "*eventually*," "*until*") and testing them against the log. AITIA-PM [70] assesses each hypothesis as a local conditional cause–effect rule (*fixed confounder*), summarized by (ϵ_{avg}), the average change in outcome probability. This approach offers clarity and local statistical significance but lacks a global process structure. Conversely, after variant–cluster disaggregation, the compendium estimates a DAG (LiNGAM/PC/GES) capturing multiple dependencies and propagations. Thus, PTL validates local relationships with limited confounding control, whereas the global DAG enables structural analysis and interventions ($\mathrm{do}(\cdot)$) beyond isolated rules [4.7, sec.3.3.2], [4.2].

(*iii*) **MAG/SMMB versus topologies without explicit latents**. MAG (*Maximal Ancestral Graph*) models causal relationships among observed variables, implicitly allowing latents represented by bidirectional edges ($\leftrightarrow$); it preserves directed acyclicity and includes all implicit adjacencies. SMMB (*Sparse Max-Min Markov Blanket*) locally identifies minimal informative parents, children, and spouses for specific explanations [69]. These techniques are effective when latent confounders are suspected but offer local scope without generating a globally oriented topology. Instead of explicitly modeling latents, the compendium mitigates confounding through variant–cluster stratification and covariate enrichment, estimating a fully oriented DAG prioritizing operational simplicity and interpretability, acknowledging potential residual latents and requiring sensitivity analyses [4.7, sec.3.3], [4.3, sec.B, sec.C].

(*iv*) **Uplift Trees versus generative structure**. Uplift trees (*models of causal-effect heterogeneity*) segment populations to maximize differences between "treated" and "control" groups via ($U(x) = \mathbb{E}[Y \mid T = 1, X = x] - \mathbb{E}[Y \mid T = 0, X = x]$), recommending subgroup-specific actions and net-value metrics [72]. Useful under ignorability, they offer operational segmentation but do not ensure global structural coherence nor clearly distinguish mediation from confounding. In the compendium, the generative structure (DAG) is first estimated to capture directions, interactions, scenario evaluation ($\mathrm{do}(\cdot)$), and explicit identification; subsequently, uplift segmentation can be integrated as a prescriptive layer coherent with this topology [4.7, sec.3.3, sec.3.3.2], [4.6].

5.3 RQ.3

How can interventions and improvement strategies aimed at mitigating temporal deviations and optimizing role dynamics in business processes be explained in causal terms?

5.3.1 Causal Explainability

In the interventionist phase (L_2), the articles included in the compendium relied on identifying ($\mathbb{E}[Y \mid \mathrm{do}(X = x)]$) from observational data, provided the DAG topology determined an adjustment set ((Z)) capable of blocking confounding between X and Y [4.2], [4.2, fig.2.7]. This decision maintained explicit and auditable assumptions: the graph indicated which variables were adjusted, why, and where, anchoring inference to clear conditions [4.7, sec.3.3.2], Chapter 2 secc(2.2.3.2). With this objective, the interventional query was rewritten using the adjustment formula $P(y \mid \mathrm{do}(x)) = \sum_z P(y \mid x, z)P(z)$. Thus, the question "what happens if we do X" was quantified (ATE/CATE) with historical data, respecting the causal mechanism that justified the adjustment [4.7, sec.3.3.2], Chapter 2 secc(2.2.3.2).

On this basis, the thesis advanced toward Explainability through counterfactual reasoning (L_3). This step required two conditions: (i) structural coherence of the DAG, and (ii) identification of the effect [4.2], [4.7, sec.3.3]. With both satisfied, the CATE ($\tau(x)$) was estimated using DML/ORF, where orthogonalization and cross-fitting reduced biases when measuring effect heterogeneity [4.6], [4.6, subsec.C]. The abduction–action–prediction scheme operated on a model with controlled confounding, such that the potential outcome (Y_x) was justified in operational terms (*for example, standardizing* complexity_level *in an engineering process*) without violating assumptions Chapter 2 secc(2.2.3.3), [4.7, sec.4.7].

In summary, L_2 provided the identifiable and auditable foundation for discussing effects under (do(·)); L_3 extended this foundation to reason about "what would have happened if..." scenarios, maintaining the connection with the causal structure. With caution, interpretation was limited to strata and conditions declared by the graph and adjustment [4.2], [4.2, fig.2.7], [4.7, sec.3.3.2], [4.6].

5.3.2 Causal Integration

The approach of the articles comprising the thesis expands the causal explainability of interventions (L_2) and counterfactuals (L_3) by technically demonstrating how effects are derived and justified compared to related works.

(***i***) In structural identification, studies based on SEM obtained PAGs that required domain knowledge to orient arcs and focused on coefficients of structural equations ([67], [68]). In contrast, the compendium comprising the thesis triangulated LiNGAM/PC/GES and obtained a fully oriented consensus DAG [4.7, sec.3.3, sec.3.3.1], [4.4, sec.5.4], [4.5, sec.4.3]; on this DAG, the back-door criterion was applied to identify ($\mathbb{E}[Y \mid do(X = x)]$) ($L_2$) and, using the same structure, estimated CATE to support the potential outcome (Y_x) (L_3) [4.2, fig.2.7].

(***ii***) In the prescriptive dimension, uplift modeling prioritized the optimization of (NetGain) through persuadable subgroups ([71], [72]); in turn, in the compendium, effects under (do(·)) were first identified using the DAG (L_2), and subsequently, CATE ($\tau(x)$) was interpreted as a potential outcome (Y_x), consistent with the intervention propagation through the network (L_3) [4.7, sec.3.3.2], [4.6]. Thus, recommendations were grounded on causal structure, not only on expected utility.

(***iii***) Regarding explanatory scope, MAG/SMMB and AITIA-PM prioritized local adjacencies (Markov Blanket) or probabilistic rules (change (ϵ_{avg})) [69], [70]; they proved effective for diagnostics (L1/L2), but their local approach limited propagation simulation. In contrast, the articles comprising the thesis used a global DAG oriented by variant to project the effects of (do(·)) through mediators (L_2), and, with the same network, simulated long-range counterfactuals (L_3) throughout the process [4.2], [4.7, fig.12], [4.7, sec.3.3].

Chapter 6
Conclusion

This thesis establishes a methodological advance in the field of Process Mining oriented towards the management of temporal deviations. By integrating Pearl's Causal Hierarchy as a guiding instrument, the research overcomes the barriers of purely observational approaches (L_1) and demonstrates the feasibility of higher levels of reasoning to explain and mitigate operational inefficiencies with engineering rigor.

The empirical evidence accumulated through the seven publications of the compendium validates the research hypothesis. It was demonstrated that the incorporation of structural causal models (L_2) and counterfactual reasoning (L_3) makes it possible to identify the root causes of inefficiencies and to estimate ex-ante the impact of interventions. Specifically, the results confirm three fundamental facts:

1. **Need for Disaggregation**: Aggregated analysis hides opposing causal mechanisms. It was verified that disaggregation by variants and temporal clustering are indispensable statistical preconditions to avoid fallacies such as Simpson's paradox in process mining.
2. **Effectiveness of Structure**: The directed causal topology (`DAG`), obtained through consensus among algorithms, offers a superior representation to correlation for distinguishing between symptoms and true causes of delays.
3. **Predictive Capacity of Interventions**: The estimation of heterogeneous effects (`CATE`) made it possible to quantify with precision how changes in roles or workloads affect cycle times before their physical implementation.

This thesis contributes to the body of knowledge in two dimensions, aligned with the specific objectives:

1. **Methodological Contribution**: A unified framework is established that elevates the discipline through Pearl's Hierarchy. The research formalized the bridge between Interpretability (level L_2) and Explainability (level L_3), explicitly integrating the Organizational Perspective to model the interaction between roles. Likewise, the introduction of Deep Learning techniques (*multiview autoencoders*) for the segmentation of latent contexts represents a significant advance in the precision of causal inference in complex environments.
2. **Practical Contribution in Applied Engineering**: An auditable decision-making tool is provided to industry. Through the `CaProM` prototype and its validation in real cases (*banking and software development*), it was demonstrated that it is possible to reduce managerial uncertainty. The system enables process managers to audit the logic of automated decisions and to justify operational interventions based on traceable mathematical evidence.

The validity of the results presents restrictions that delimit the scope of the proposed solution. First, the reliability of causal graphs depends intrinsically on the completeness of event logs; the presence of latent confounders not instrumented in the logs may introduce bias into effect estimation. Second, although robust algorithms were used (`LiNGAM`, `PC`, `GES`), the computational cost of causal discovery scales exponentially with process dimensionality, which may require optimizations for real-time deployments. Finally, the interpretation of counterfactual scenarios requires validation by domain experts to ensure the operational feasibility of interventions.

This research opens new lines of development for intelligent process engineering, projected along three fundamental axes:

1. **Self-Adaptive Systems (`MAPE-K`)**: The integration of the causal framework within the `MAPE-K` cycle (*Monitor, Analyze, Plan, Execute, Knowledge*) is proposed. This will allow the current system to evolve towards an autonomous architecture capable of monitoring deviations, analyzing root causes through graphs, and executing mitigation plans in real time, closing the adaptive control loop.

2. **Causal Reinforcement Learning (`CRL`)**: Future research may extend prescriptive capabilities through Causal Reinforcement Learning. Unlike traditional `RL`, this approach will allow agents to learn optimal intervention policies by distinguishing between spurious correlations and real causal mechanisms, accelerating convergence and robustness in stochastic environments.

3. **Causal Transportability**: Finally, it will be possible to explore the transportability of the discovered models. The goal is to transfer causal structures and effects learned in a specific variant or context to new operational environments, determining which mechanisms are invariant and reducing the need to retrain models from scratch.

Ultimately, this thesis closes the gap between data observation and intelligent action in process management by deploying the Causal Hierarchy as the missing methodological link. In doing so, it provides a rigorous solution for the governance of temporal deviations, transforming a complex operational problem into a systematic source of organizational learning.

Bibliography

[1] W.M.P. van der Aalst, A. Adriansyah, A.K. Alves de Medeiros, F. Arcieri, T. Baier, T. Blickle, J.C. Bose, P. van den Brand, R. Brandtjen, J. Buijs, et al.: Process mining manifesto. In: International Conference on Business Process Management, pp. 169–194. Springer, Cham (2011). https://doi.org/10.1007/978-3-642-28108-2_19

[2] Mahdia, A., Boukhedouma, S., Alimazghi, Z.: Impact of strategic alignment on company performance: An approach based on performance indicators system design. In: 2020 International Conference on Advanced Computer Science and Information Systems (ICACSIS) (2020).

[3] Sontheim, J., Richter, F., Seidl, T.: Temporal deviations on event sequences. CEUR Workshop Proceedings, 2454 (2019).

[4] Janicki, J., Wójcik, E.: Tools for analysis of business processes – a comparative analysis. Journal of Computer Sciences Institute, 18, 152–159 (2021). https://doi.org/10.35784/JCSI.2652

[5] França, J.B.S., Borges, M.R.S.: Systematizing the impacts projection of complex decisions in work groups. SN Applied Sciences, 2(7), 1287 (2020). https://doi.org/10.1007/s42452-020-3086-4

[6] Leemans, S.J.J., McGree, J., Polyvyanyy, A., ter Hofstede, A.H.M.: Statistical tests and association measures for business processes. IEEE Transactions on Knowledge and Data Engineering, 35, 7497–7511 (2023). https://doi.org/10.1109/TKDE.2022.3197840

[7] Vukovic, M., Thalmann, S.: Causal discovery in manufacturing: A structured literature review. Journal of Manufacturing and Materials Processing, 6(1), 10 (2022). https://doi.org/10.3390/jmmp6010010

[8] Mehdouani, K., Missaoui, N., Ghannouchi, S.A.: An approach for business process improvement based on simulation technique. Procedia Computer Science, 164, 225–232 (2019).

[9] Botelho, J.G.S., Santos, E.A.P., Choueiri, A.C., Pécora Junior, J.E.: Remaining time prediction in manufacturing systems: An approach based on ML and process mining. Procedia CIRP, 132, 165–170 (2025). https://doi.org/10.1016/j.procir.2025.01.028

[10] Wuennenberg, M., Wegerich, B., Fottner, J.: Optimization of internal logistics using a combined BPMN and simulation approach. In: Proceedings of the 2022 Winter Simulation Conference, pp. 13–19 (2022). https://doi.org/10.7148/2022-0013

[11] Gershman, S.J., Ullman, T.D.: Causal implicatures from correlational statements. PLOS ONE, 18(5), e0286067 (2023). https://doi.org/10.1371/journal.pone.0286067

[12] Miklin, N., Gachechiladze, M., Moreno, G., Chaves, R.: Causal inference with imperfect instrumental variables. Journal of Causal Inference, 10(1), 45–63 (2022). https://doi.org/10.1515/jci-2021-0065

[13] Zimmermann, O., Luban, K., Stocker, M., Bernard, G.: Continuous process model refinement from business vision to event simulation and software automation: Bridging gaps between stakeholder communities, practices, notations, and tools. In: 2022 IEEE/ACM International Workshop on Software-Intensive Business (IWSiB), pp. 59–66. IEEE (2022). https://doi.org/10.1145/3524614.3528631

[14] Zanga, A., Ozkirimli, E., Stella, F.: A survey on causal discovery: Theory and practice. International Journal of Approximate Reasoning, 151, 101–129 (2022). https://doi.org/10.1016/j.ijar.2022.09.004

[15] Yao, L., Chu, Z., Li, S., Li, Y., Gao, J., Zhang, A.: A survey on causal inference. ACM Transactions on Knowledge Discovery from Data, 15(5), 1–37 (2021). https://doi.org/10.1145/3444944

[16] Ahmed, R., Faizan, M., Burney, A.I.: Process mining in data science: A literature review. In: 2019 International Conference on Advances in the Emerging Computing Technologies (AECT), pp. 1–6. IEEE (2019). https://doi.org/10.1109/MACS48846.2019.9024803

[17] Mann, A., Watt, G., Matthews, P.: Pull and push. In: The Innovative CIO, pp. 65–77. Apress, Berkeley, CA (2013). https://doi.org/10.1007/978-3-031-05760-1_5

[18] Badakhshan, P., Wurm, B., Grisold, T., Geyer-Klingeberg, J., Mendling, J., vom Brocke, J.: Creating business value with process mining. Journal of Strategic Information Systems, 31(3), 101745 (2022). https://doi.org/10.1016/j.jsis.2022.101745

[19] Witmer, J.: Simpson's paradox, visual displays, and causal diagrams. American Mathematical Monthly, 128(7), 598–610 (2021). https://doi.org/10.1080/00029890.2021.1932237

[20] Chou, Y.L., Moreira, C., Bruza, P., Ouyang, C., Jorge, J.: Counterfactuals and causability in explainable artificial intelligence: Theory, algorithms, and applications. Information Fusion, 81, 59–83 (2022). https://doi.org/10.1016/j.inffus.2021.11.003

[21] Taymouri, F., La Rosa, M., Dumas, M., Maggi, F.M.: Business process variant analysis: Survey and classification. Knowledge-Based Systems, 211, 106557 (2021). https://doi.org/10.1016/j.knosys.2020.106557

[22] Parida, P.K.: Causality and its applications. (2021).

[23] Burattin, A., Conti, M., Turato, D.: Toward an anonymous process mining. In: 2015 3rd International Conference on Future Internet of Things and Cloud, pp. 723–728. IEEE (2015). https://doi.org/10.1109/FICLOUD.2015.9

[24] Meddah, I.H., Guerroudji, F.: Business process discovery using process mining techniques and distributed framework. International Journal of Cloud Applications and Computing (IJCAC), 12(1), 1–17 (2022). https://doi.org/10.4018/ijcac.300772

[25] AlShathry, O.: Process mining as a business process discovery technique. Journal of Computer Sciences and Applications, 4(3), 83–87 (2016). https://doi.org/10.4172/2324-9307.1000141

[26] Carmona, J., van Dongen, B., Weidlich, M.: Conformance checking: Foundations, milestones and challenges. In: van der Aalst, W.M.P., Carmona, J. (eds.) Process Mining Handbook. Lecture Notes in Business Information Processing, vol. 448, pp. 79–115. Springer, Cham (2022). https://doi.org/10.1007/978-3-031-08848-3_5

[27] Rossi, P.H., Lipsey, M.W., Henry, G.T.: Evaluation: A systematic approach. Sage Publications, Thousand Oaks, CA (2018).

[28] Escobar-Sierra, M., Lara-Valencia, L.A., ValenciaDeLara, P.: 'Step-by-step' method to conduct applied research in organizational engineering and business management. Culture and Education, 33(1), 28–77 (2021). https://doi.org/10.1080/11356405.2020.1859735

[29] Kubrak, K., Milani, F., Nolte, A.: Process Mining for Process Improvement - An Evaluation of Analysis Practices. In: Guizzardi, R., Ralyté, J., Franch, X. (eds.) Research Challenges in Information Science (RCIS 2022). Lecture Notes in Business Information Processing, vol. 446, pp. 214–230. Springer, Cham (2022). https://doi.org/10.1007/978-3-031-05760-1_13

[30] van Eck, M.L., Lu, X., Leemans, S.J.J., van der Aalst, W.M.P.: PM: A Process Mining Project Methodology. In: Zdravkovic, J., Kirikova, M., Johannesson, P. (eds.) Advanced Information Systems Engineering (CAiSE 2015). Lecture Notes in Computer Science, vol. 9097, pp. 297–313. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-19069-3_19

[31] Rossi, P.H., Lipsey, M.W., Henry, G.T.: Evaluation: A systematic approach. Sage Publications, Thousand Oaks, CA (2018).

[32] Escobar-Sierra, M., Lara-Valencia, L.A., Valencia De Lara, P.: 'Step-by-step' method to conduct applied research in organizational engineering and business management. Culture and Education (2021). https://doi.org/10.1080/11356405.2020.1859735

[33] Kubrak, K., Milani, F., Nolte, A.: Minería de procesos para la mejora de procesos: una evaluación de las prácticas de análisis. En: Guizzardi, R., Ralyté, J., Franch, X. (eds.) Desafíos de la investigación en la ciencia de la información. Lecture Notes in Business Information Processing, vol. 446, pp. 185–201. Springer, Cham (2022). https://doi.org/10.1007/978-3-031-05760-1_13

[34] van Eck, M.L., Lu, X., Leemans, S.J.J., van der Aalst, W.M.P.: PM: A process mining project methodology. In: Zdravkovic, J., Kirikova, M., Johannesson, P. (eds.) Advanced Information Systems Engineering. CAiSE 2015. Lecture Notes in Computer Science, vol. 9097. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-19069-3_19

[35] Kalenkova, A.A., van der Aalst, W.M.P., Lomazova, I.A., et al.: Process mining using BPMN: Relating event logs and process models. Software and Systems Modeling, 16, 1019–1048 (2017). https://doi.org/10.1007/s10270-015-0502-0

[36] van der Aalst, W.M.P.: Foundations of process discovery. In: van der Aalst, W.M.P., Carmona, J. (eds.) Process Mining Handbook. Lecture Notes in Business Information Processing, vol. 448. Springer, Cham (2022). https://doi.org/10.1007/978-3-031-08848-3_2

[37] van der Aalst, W.M.P.: Decomposing Petri nets for process mining: A generic approach. Distributed and Parallel Databases, 31, 471–507 (2013). https://doi.org/10.1007/s10619-013-7127-5

[38] Augusto, A., Carmona, J., Verbeek, E.: Advanced process discovery techniques. In: van der Aalst, W.M.P., Carmona, J. (eds.) Process Mining Handbook. Lecture Notes in Business Information Processing, vol. 448. Springer, Cham (2022). https://doi.org/10.1007/978-3-031-08848-3_3

[39] van der Aalst, W.M.P.: Process mining: A 360 degree overview. In: van der Aalst, W.M.P., Carmona, J. (eds.) Process Mining Handbook. Lecture Notes in Business Information Processing, vol. 448. Springer, Cham (2022). https://doi.org/10.1007/978-3-031-08848-3_1

[40] De Weerdt, J., Wynn, M.T.: Foundations of process event data. In: van der Aalst, W.M.P., Carmona, J. (eds.) Process Mining Handbook. Lecture Notes in Business Information Processing, vol. 448. Springer, Cham (2022). https://doi.org/10.1007/978-3-031-08848-3_6

[41] Yao, L., Chu, Z., Li, S., Li, Y, Gao, J., Zhang, A.: A survey on causal inference. ACM Transactions on Knowledge Discovery from Data, 15(5), Article 74 (2021). https://doi.org/10.1145/3444944

[42] Kuang, K., Li, L., Geng, Z., Xu, L., Zhang, K., Liao, B., Huang, H., Ding, P., Miao, W., Jiang, Z.: Causal inference. Engineering, 6, 253–263 (2020). https://doi.org/10.1016/j.eng.2019.08.016

[43] Johansson, L.G., et al.: Causal explanations. In: A Primer to Causal Reasoning About a Complex World. SpringerBriefs in Philosophy. Springer, Cham (2024). https://doi.org/10.1007/978-3-031-59135-8_8

[44] Bareinboim, E.: Causal Artificial Intelligence: A Roadmap for Building Causally Intelligent Systems. 2025. https://causalai-book.net.

[45] Rubin, D.B.: Causal inference using potential outcomes: Design, modeling, decisions. Journal of the American Statistical Association, 100(469), 322–331 (2005).

[46] Wenjuan, W., Lu, F., Chunchen, L.: Mixed causal structure discovery with application to prescriptive pricing. In: Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence (IJCAI-18), pp. 5126–5134 (2018). https://doi.org/10.24963/ijcai.2018/711

[47] Zheng, M., Marsh, J.K., Nickerson, J.V., Kleinberg, S.: How causal information affects decisions. Cognitive Research: Principles and Implications, 5(1), 6 (2020). https://doi.org/10.1186/s41235-020-0206-z

[48] Bulbulia, J., Schjoedt, U., Shaver, J.H., Sosis, R., Wildman, W.J.: Causal inference in regression: Advice to authors. Religion, Brain & Behavior, 11(4), 353–360 (2021). https://doi.org/10.1080/2153599X.2021.2001259

[49] Austin, P.C.: An introduction to propensity score methods for reducing the effects of confounding in observational studies. Multivariate Behavioral Research, 46(3), 399–424 (2011).

[50] Wang, Y., Luo, Z.: Causal discovery and causal inference based counterfactual fairness in machine learning. In: ICASSP 2023 – 2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Rhodes Island, Greece, pp. 1–5 (2023). https://doi.org/10.1109/ICASSP49357.2023.10095194

[51] Rogge-Solti, A., Kasneci, G.: Temporal anomaly detection in business processes. In: Sadiq, S., Soffer, P., Völzer, H. (eds.) Business Process Management. BPM 2014. Lecture Notes in Computer Science, vol. 8659, pp. 234–249. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-10172-9_15

[52] Pereira, J.L., Varajão, J.: The temporal dimension of business processes – Dealing with time constraints. Procedia Computer Science, 121, 1034–1038 (2017). https://doi.org/10.1016/j.procs.2017.11.133

[53] Senderovich, A., Weidlich, M., Gal, A.: Temporal network representation of event logs for improved performance modelling in business processes. In: Carmona, J., Engels, G., Kumar, A. (eds.) Business Process Management. BPM 2017. Lecture Notes in Computer Science, vol. 10445. Springer, Cham (2017).

[54] Böhmer, K., Rinderle-Ma, S.: Multi instance anomaly detection in business process executions. In: Business Process Management: 15th International Conference, BPM 2017, Barcelona, Spain, September 10–15, 2017, Proceedings 15, pp. 77–93. Springer International Publishing (2017).

[55] Sontheim, J., Richter, F., Seidl, T.: Temporal deviations on event sequences. CEUR Workshop Proceedings, 2454 (2019).

[56] Saralaya, S., D'Souza, R.G., Saralaya, V.: Temporal impact analysis and adaptation for service-based systems. International Journal of Information and Communication Technology, 14(4), 403–426 (2019).

[57] Richter, F., Seidl, T.: Looking into the TESSERACT: Time-drifts in event streams using series of evolving rolling averages of completion times. Information Systems, 84, 265–282 (2019). https://doi.org/10.1016/j.is.2018.11.003

[58] Mavroudopoulos, I., Gounaris, A.: Detecting temporal anomalies in business processes using distance-based methods. In: Discovery Science: 23rd International Conference, DS

2020, Thessaloniki, Greece, October 19–21, 2020, Proceedings 23, pp. 615–629. Springer International Publishing (2020).

[59] Chapela-Campa, D., Dumas, M.: Enhancing business process simulation models with extraneous activity delays. Information Systems, 122, 102346 (2024). https://doi.org/10.1016/j.is.2024.102346

[60] Magallanes, J., Stone, T., Morris, P.D., Mason, S., Wood, S., Villa-Uriol, M.C.: Sequen-C: A multilevel overview of temporal event sequences. IEEE Transactions on Visualization and Computer Graphics, 28(1), 901–911 (2022).

[61] Zhao, D., Zhou, Z.: Anomaly and interestingness detection in timed hierarchical business processes. IEEE Transactions on Engineering Management (2022). https://doi.org/10.1109/TEM.2022.3182413

[62] Richter, F., Lu, Y., Zellner, L., Sontheim, J., Seidl, T.: TOAD: Trace ordering for anomaly detection. In: 2020 2nd International Conference on Process Mining (ICPM), pp. 169–176. IEEE (2020).

[63] Ouarhim, A., Baina, K., Elbhiri, B.: Towards a continuous temporal improvement approach for real-time business processes. International Journal of Advanced Computer Science and Applications (IJACSA), 15(1) (2024).

[64] Cohen, I.: Data-driven project planning: An integrated network learning, process mining, and constraint relaxation approach in favor of scheduling recurring projects. IEEE Transactions on Engineering Management, 71, 7719–7729 (2024). https://doi.org/10.1109/TEM.2024.3382727

[65] Zhao, D., Zhou, Z., Wang, Y., Gaaloul, W.: Detecting temporal anomaly and interestingness in timed business process models. In: 2020 IEEE International Conference on Web Services (ICWS), Beijing, China, pp. 418–422 (2020). https://doi.org/10.1109/ICWS49710.2020.00061

[66] Mavroudopoulos, I., Gounaris, A.: A comparison of proximity-based methods for detecting temporal anomalies in business processes. Machine Learning, 112, 4101–4128 (2023). https://doi.org/10.1007/s10994-022-06152-5

[67] Qafari, M.S., van der Aalst, W.M.P.: Root cause analysis in process mining using structural equation models. In: Del Río Ortega, A., Leopold, H., Santoro, F.M. (eds.) Business Process Management Workshops. BPM 2020. Lecture Notes in Business Information Processing, vol. 397, pp. 155–167. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-66498-5_12

[68] Qafari, M.S., van der Aalst, W.M.P.: Feature recommendation for structural equation model discovery in process mining. Progress in Artificial Intelligence (2022). https://doi.org/10.1007/s13748-022-00282-6

[69] Luo, Q., Zhang, L., Xing, Z., Xia, H., Chen, Z.-X.: Causal discovery of flight service process based on event sequence. Journal of Advanced Transportation, 2021, 2869521, 17 pages (2021). https://doi.org/10.1155/2021/2869521

[70] Van Houdt, G., Depaire, B., Martin, N.: Root cause analysis in process mining with probabilistic temporal logic. In: Munoz-Gama, J., Lu, X. (eds.) Process Mining Workshops. ICPM 2021. Lecture Notes in Business Information Processing, vol. 433, pp. 73–84. Springer, Cham (2022).

[71] Bozorgi, Z.D., Dumas, M., La Rosa, M., Polyvyanyy, A., Shoush, M., Teinemaa, I.: Learning when to treat business processes: Prescriptive process monitoring with causal inference and reinforcement learning. In: Business Process Management Forum, pp. 285–301. Springer (2023). https://doi.org/10.1007/978-3-031-34560-9_22

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

[73] Fournier, F., Limonad, L., Skarbovsky, I., et al.: The WHY in business processes: Discovery of causal execution dependencies. Künstliche Intelligenz (2025). https://doi.org/10.1007/s13218-024-00883-4

[74] Eshuis, R., Genga, L.: Intervention strategies for improving business process outcomes: A causal rule mining approach. Expert Systems with Applications, 284, 127745 (2025). https://doi.org/10.1016/j.eswa.2025.127745

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

[76] Hompes, B.F., Maaradji, A., La Rosa, M., Dumas, M., Buijs, J.C., van der Aalst, W.M.P.: Discovering causal factors explaining business process performance variation. In: International Conference on Advanced Information Systems Engineering, pp. 177–192. Springer International Publishing, Cham (2017).

[77] Panpanich, P., Porouhan, P., Premchaiswadi, W.: Analysis of handover of work in call center using social network process mining technique. In: 2015 13th International Conference on ICT and Knowledge Engineering (ICT & Knowledge Engineering 2015), Bangkok, Thailand, pp. 97–104 (2015). https://doi.org/10.1109/ICTKE.2015.7368478

Appendix
Collaborative Publications Not Included in the Compendium

This chapter brings together a set of publications carried out in collaboration with researchers from various institutions. Although external to the core compendium, these works address engineering and data science problems in different application domains. Their inclusion documents the research trajectory developed in parallel, as well as participation in collaborative initiatives and the sustained scientific production during the doctoral period.

Publication Summary

Title: *Multi-View Spectral Clustering for Monolith to Microservices Migration*

Authors: Gonzalo Rivera-Lazo, **Fernando Montoya** and Hernán Astudillo

Conference: 28th Ibero-American Conference on Software Engineering (CIbSE)

Publisher: Digital Library maintained by the Brazilian Computer Society (SBC)

Year: 2025

Doi: https://doi.org/10.5753/cibse.2025.35321

Publication Summary

Title: *Crosscutting Teaching Strategies for Software Development Teams: Insights from Kolb's Inventory*

Authors: Mauricio Hidalgo, Kattia Rodriguez, Laura M. Castro, **Fernando Montoya** and Hernán Astudillo

Journal: CLEI *Electronic Journal*

Year: 2025

Doi: https://doi.org/10.19153/cleiej.28.5.5

Publication Summary

Title:	*Enhancing Software Requirements Education Through Active Learning: A Pilot Study on Role-Playing and Real-World Simulations*
Authors:	Mauricio Hidalgo, Laura M. Castro, Hernán Astudillo, **Fernando Montoya** and Manuel Goyo
Conference:	51st Latin American Computer Conference (CLEI)
Publisher:	IEEE *Xplore*
Year:	2025
Doi:	https://doi.org/10.1109/CLEI67442.2025.11420808

Publication Summary

Title:	*Technical Debt in Transfer Learning: A Rapid Review of* IEEE *Xplore for a Call to Action*
Authors:	Mauricio Hidalgo, **Fernando Montoya**, Cristian Beltrán and Hernán Astudillo
Conference:	IEEE Baja California (BCC)
Publisher:	IEEE *Xplore*
Year:	2025

Publication Summary

Title:	*Improving Early and Differential Dengue Diagnosis with Gradient Boosting: A Comparative Approach Based on Clinical Data from Sucre, Colombia*
Authors:	Humberto Marbello-Peña, Wilson Arrubla-Hoyos, Jaider Caldera-Nadaff, **Fernando Montoya** and Zurisaddai Severiche-Maury
Conference:	IEEE Baja California (BCC)
Publisher:	IEEE *Xplore*
Year:	2025

Publication Summary

Title:	*Forecasting Critical Energy Consumption Hours per Device in Residential Environments Using* HEMS *and* LSTM
Authors:	Zurisaddai Severiche-Maury, Wilson Arrubla-Hoyos, César Viloria-Núñez, **Fernando Montoya** and Mauro González-Sierra
Conference:	IEEE Baja California (BCC)
Publisher:	IEEE *Xplore*
Year:	2025